2017

Investigate Genomic 3D Structure Using Deep Neural Network

Yan Zhang
University of South Carolina

Investigate genomic 3D structure using deep Neural network

by

Yan Zhang

Bachelor of Science
Shandong University, 2009

Master of Science
The University of Alabama, 2011

Submitted in Partial Fulfillment of the Requirements

for the Degree of Doctor of Philosophy in

Computer Science

College of Engineering and Computing

University of South Carolina

2017

Accepted by:

Jijun Tang, Major Professor

Gabriel A. Terejanu, Committee Member

John R. Rose, Committee Member

Yan Tong, Committee Member

Jiajia Zhang, Committee Member

Cheryl L. Addy, Vice Provost and Dean of the Graduate School

Acknowledgments

First and foremost I wish to gratefully and sincerely thank my advisor Dr. Jijun Tang for his guidance, understanding, patience. He encourages me to discover my real passion in research, and also provided continuous guidance, understanding, and patience along the way. Also, his mentorship was paramount in providing a well-rounded experience consistent my long-term career goals.

I would like to thank all my committee members: Dr. Rose, Dr. Terejanu, Dr. Tong, and Dr. Zhang for their willingnesses to serve on my dissertation committee and their valuable comments.

I would also like to thank Dr. Feng Yue from Penn. State Univ. for his guidance and assistance in research topic as well as the advice in writing. I am very grateful for the friendship and cooperation with him.

I would like to acknowledge my wife, my parents and my parents-in-law for their support and encouragement to finish my Ph.D.

Abstract

The 3D structures of the chromosomes play fundamental roles in essential cellular functions, e.g., gene regulation, gene expression, evolution and Hi-C technique provides the interaction density between loci on chromosomes. In this dissertation, we developed multiple algorithms, focusing the deep learning approach, to study the Hi-C datasets and the genomic 3D structures.

Building 3D structure of the genome one of the most critical purpose of the Hi-C technique. Recently, several approaches have been developed to reconstruct the 3D model of the chromosomes from HiC data. However, all of the methods are based on a particular mathematical model and lack of flexibility for new development.We introduce a novel approach using the genetic algorithm. Our approach is flexible to accept any mathematical models to build a 3D chromosomal structure. Also, our approach outperforms current techniques in accuracy.

Although an increasing number of Hi-C datasets have been generated in a variety of tissue/cell types, Due to high sequencing cost, the resolution of most Hi-C datasets are coarse and cannot be used to infer important biological functions (e.g., enhancer-promoter interactions, and link disease-related non-coding variants to their target genes). To address this challenge, we develop HiCPlus, a computational approach based on deep convolutional neural network, to infer high-resolution Hi-C interaction matrices from low-resolution Hi-C data. Through extensive testing, we demonstrate that HiCPlus can impute interaction matrices highly similar to original ones while using only as few as 1/16 of the total sequencing reads. We observe that Hi-C interaction matrix contains unique local features that are consistent across different cell

types, and such features can be effectively captured by the deep learning framework. We further apply HiCPlus to enhance and expand the usability of Hi-C datasets in a variety of tissue and cell types. In summary, our work not only provides a framework to generate high-resolution Hi-C matrix with a fraction of the sequencing cost but also reveals features underlying the formation of 3D chromatin interactions.

The noise level in the Hi-C is high, and the structure of the noise is complicated. Also, even under most strict experimental conditions, the absolute noise-free Hi-C data still cannot be obtained. We proposed a novel approach to learn a denoising network without clean data. Our approach employs Siamese structure, utilizing two replicates of the same experimental settings to train the model; the resulting model can then be applied to datasets where only one replicate is available. We applied our new approach to enhance Hi-C data, an important type of data in exploring three-dimensional genomic structures. The results prove that the model trained by our method significantly reduce the noise level in Hi-C data.

In the past few years, we have seen an explosion of Hi-C data in a variety of cell/tissue types. While these publicly available data presents an unprecedented opportunity to interrogate chromosomal architecture, how to quantitatively compare Hi-C data from different tissues and identify tissue-specific chromatin interactions remains challenging. We developed HiCComp, a comprehensive framework for comparing Hi-C data. HiCComp utilizes convolutional neural networks to extract key features in Hi-C interaction matrices in a fully automatic way. The core component of HiCComp is a triplet network, which contains three identical convolutional neural networks with shared parameters. The inputs to our network are three Hi-C matrices: two of them are biological replicates from the same cell type, and the third one is from another cell type. The HiCComp network takes advantages of the two biological replicates to estimate the natural variation in the experiments and further use it to identify significant variations between Hi-C matrices from different cell

types. Furthermore, we incorporate systematic occluding method into our framework so that we can identify the dynamic interaction regions from Hi-C maps. Finally, we show that the dynamic regions between two cell types are enriched for transcription factor binding sites and histone modifications that are associated with cis-regulatory functions, suggesting these variations in 3D genome structure are potentially gene regulatory events.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

Epigenomics (also know as "epigenetics") studies the chromosomal states variations which are associated with cellular functions but do not involve the modification of DNA sequences. All types of cells in the body essentially share the same genome from the single fertilized egg, and epigenomic factors are the interpretation of the specific identities of different cell types (Consortium et al. 2012). The research in epigenomics contributes to answer a series of fundamental biological functions such as:

- What are the key regulatory elements determine the gene expressions in different tissues/cell types?
- What is the control mechanism in the development of embryos?
- What are factors contributing to the cellular functions?

Because of advances in high-throughput experimental techniques, a broad range of epigenomic states can be detected including modifications of DNA molecules, DNA-protein bindings, and genomic 3D conformations. In this dissertation, we will focus on the genomic 3D conformation, and its connection to other epigenomic states.

Due to the importance of epigenomics, a large volume of epigenomic data has been published, and several projects, including NIH Roadmap (`www.roadmapepigenomics.org/`) and ENCODE (`https://www.encodeproject.org/`) are aiming to systemically perform experiments to measure epigenomic markers in a varieties of cell lines

and tissue types. Therefore, it is urgent to develop algorithms and computational tools to analyze epigenomic data from experiments.

Deep learning is a rapidly developing technique in recent years and has achieved great success in many domains, including computer vision, natural language process and even playing Go (LeCun, Bengio, and G. Hinton 2015; Goodfellow, Bengio, and Courville 2016; Silver et al. 2016). Unlike conventional machine-learning techniques, such as logistic regression or SVM, deep learning allows the model to directly process the natural data in their raw form (e.g. the pixels on an image or the nucleotide on the genome) without much domain expertise and feature engineering process.

Epigenomics is an emerging field, yet the methods to analyze high-throughput epigenomic data is very limited. In this dissertation, we employ deep learning techniques to develop algorithms for the analysis of epigenomic data, focusing on data related to genomic 3D structures.

1.1 Biological background

1.1.1 Epigenomics

As we discussed previously, different types of cells have the same set of genome. Cell types utilized in the epigenomic research are classified as two types: tissues and cell lines. Tissues (e.g. brain, muscle, liver) are real human cells, while cell lines (e.g. GM12878, IMR90, K562) are the standard cells under treatment in vitro. Since the source of tissues is limited and the epigenomic state of tissues may vary across different individuals, many research employs cell lines to investigate epigenomic properties.

Traditionally, the difference across cell types can only be observed by their functions with limited and incomplete data available (as shown in the brown part In Fig. 1.5). As the development of high-throughput sequencing techniques, the detection of chromosomal global state has become possible. Currently, most of the epigenomic studies are focused on the following two aspect:

- **The event along the DNA sequence on the genome**: The events include proteins binding on the DNA string and the modification of the histone, which the DNA string wrapping around.

- **The spatial conformation of the genome**: The spatial structure of the chromosome plays a significant role in gene functions, e.g., gene regulation, gene expression (Misteli 2007; T. Cremer and C. Cremer 2001; Sexton et al. 2007; P. Fraser and W. Bickmore 2007; W. A. Bickmore 2013) and cell differentiation (J. R. Dixon et al. 2015). The 3D structure of chromosomes is very complicated as we need to pack 3 billion base pairs into a space no bigger than 10 microns in diameter.

1.1.2 Experimental techniques to obtain high-throughput epigenomic data

In the epigenomic field, the key part is to determine the locations of the epigenomic events. Thanks to the development of DNA sequencing technologies in past several decades, the reference genome of many species have already been built. Therefore, the location of the event can be represented as the location on the DNA of the reference genome (e.g. chromosome 1: 1000000-1000200).

The other key technique contributes to the advances in epigenomic studies is short DNA segment sequencing technique, which enables us to sequence a large volume short DNA segments (also names as "reads") in relative cheap prices. Coupling with the progress in the sequencing mapping algorithm (H. Li and Durbin 2009), we can locate the original locations of these short DNA segments (Fig. 1.1).

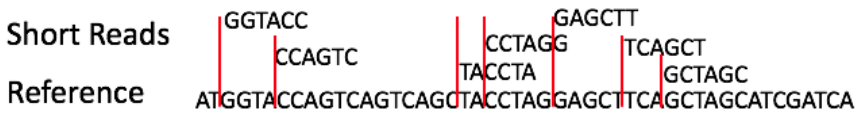


Figure 1.1: Determine the location of the short DNA sequencing reads on the reference genome.

Chip-seq Chip-seq (Johnson et al. 2007) is a class of techniques to detect the epigenomic state along the DNA sequence. To sequence the DNA reads, the entire DNA string is cut into small segments, and traces of the epigenomic states on the location(e.g., protein binding) can be catches by the experiment techniques named chromatin immunoprecipitation(ChIP), which can precipitate the DNA segments with a specific state(e.g., binding to a certain kind of protein). By sequencing the precipitated DNA segments only, we can obtain the locations of the distribution of the state on the entire genome and the process of Chip-seq is shown in Fig. 1.2.

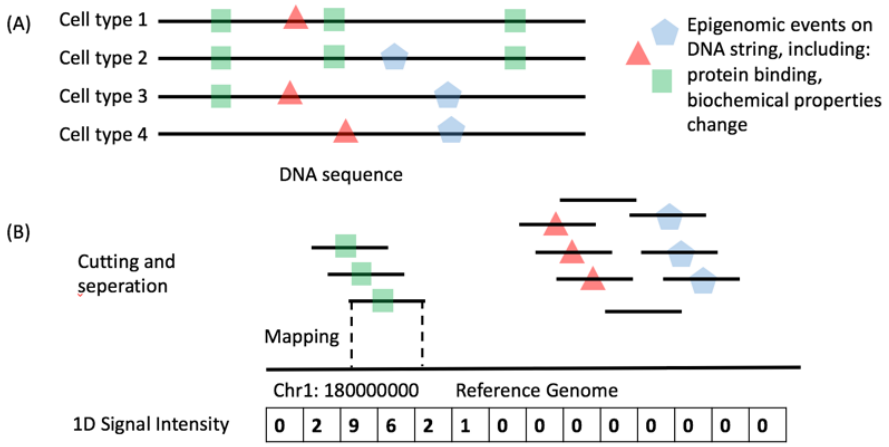
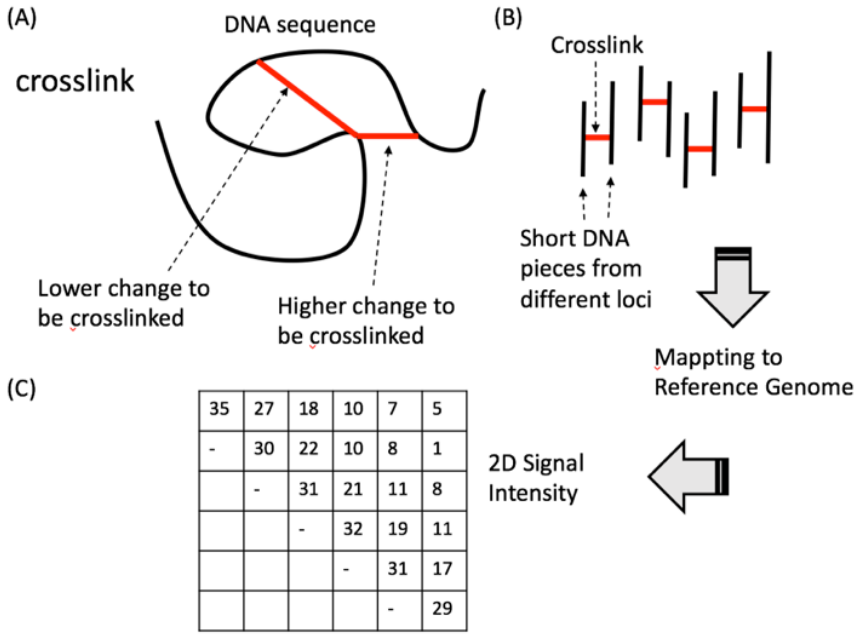


Figure 1.2: The mechanism of the Chip-seq to capture the epigenomic event on the DNA sequence.

Figure 1.2a shows the epigenomic state of the four cell types. Although these four cell types share the same set of DNA sequence, they have different epigenomic events at different locations, and the events may include (but not limited to) protein bindings and histone modifications. To determine the location of such events, the DNA string is cut to small segments; then the segments with specific type of event are separated from other segments. The filtered small DNA segments of the DNA string are sequenced and mapped to the reference genome to obtain the coordination of these pieces on the reference genome . Since the experiment is performed on a group of cells, each location may have multiple DNA sequencing reads as shown in

Fig. 1.2b.



35	27	18	10	7	5
-	30	22	10	8	1
	-	31	21	11	8
		-	32	19	11
			-	31	17
				-	29

Figure 1.3: The mechanism of the Hi-C to capture the crosslink from the 3D structure of the chromosomes.

Hi-C Hi-C is a technique to detect the interaction between two loci on the genome. As genomes are packed in a very small space, it is still impossible to directly observe their structure through equipment such as microscopes (Marti-Renom and Mirny 2011). Consequently, Hi-C is the major approach to detect the 3D structure of the genome.

Similar to Chip-seq, Hi-C also employs the sequencing and mapping technique to determine the original locations of DNA segments. The difference between Hi-C and Chip-Seq is the process to generate the short DNA segments. The following are the general steps for the Hi-C experiment.

1. Cross-linking the DNA string on the genome. In this step, the formaldehyde is employed to bind two loci on DNA string, and the loci with shorter distance in 3D space has higher chance to be cross-linked (Fig. 1.3a).

2. The DNA string is cut into small pieces, and only the cross-linked DNA segments are kept in the filter process.

3. The obtained DNA segments are sequenced and mapped to the reference genome (Fig. 1.3b)

4. The number of the cross-linked DNA segments between each pair of loci are calculated to obtain the Hi-C signal intensities. The Hi-C experiment produces a 2D matrix and the bins on the matrix are fixed-size, non-overlapping and continuous windows along the DNA sequence Fig. 1.3c.

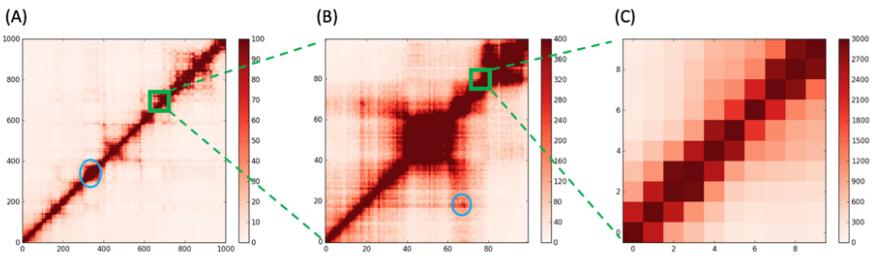


Figure 1.4: The Hi-C data represented by heatmap.

For the Hi-C data, each bin on the matrix of Fig. 1.3c is from 5k bp to 100k bp and the size of the matrix is millions by millions for the entire genome. Therefore, it is impossible to present the Hi-C data matrix using the exact number of reads. A common way to represent the Hi-C matrix is to use heatmap as shown in Fig. 1.4. In Fig. 1.4, each bin on the matrix represents a 25k bp region, so the Fig. 1.4a shows the Hi-C data from 25M bp, which covers one third of chromosome 18, the smallest chromosomes in human.

ChIA-PET Besides 3C-based methods, Chromatin Interaction Analysis by Paired-End Tag Sequencing (ChIA-PET) (Fullwood et al. 2009) is another insightful approach to detect genome-wide interaction intensities. ChIA-PET combines chromatin

immunoprecipitation (ChIP)-based enrichment (Johnson et al. 2007), chromatin proximity ligation and high-throughput pair-end sequencing techniques to detect the interaction intensities between several protein-binding regions on the genome (Fullwood et al. 2009; Bonev and Cavalli 2016; Tang et al. 2015). Compared with the Hi-C technique, ChIA-PET only detects the region where the DNA is bound with target proteins rather than the entire genome. Therefore, ChIA-PET can only give the interaction intensities between the loci from a subset of the genomic region, making ChIA-PET more efficiency in detecting a specific kind of the genomic interactions.

In Fig. 1.5, we summarize the available high-throughput epigenomic data sets and their relationship to the biological meanings. The blue region covers the hidden state of the epigenomic state of a cell, and most of the factors and their relationship are still largely unknown. Traditionally, some of the functions of the cell can be observed and validated by experiments but most of such observations are very limited and incomplete. As the development of the high-throughput sequencing technique, the detection of the chromosomal global state become possible, and the most used techniques in the epigenomics are named Chip-Seq, ChIA-PET and Hi-C. The Chip-seq technique reflected the 1D characters along the DNA sequencing, such as the protein binding and the biochemical properties changes on the DNA string. Hi-C, which are the core data sets in this dissertation, detect the interaction between two loci on the DNA string and provide the information of the 3D structures. ChIA-PET can be considered as the combination of Hi-C and Chip-seq.

1.2 Deep learning

1.2.1 The development of Deep Learning

The deep learning is also called deep neural networks. The concept of the neuron, which is the basic unit in any deep neural network, was raised by McCulloch at 1943 based on the model of brain function (McCulloch and Pitts 1943). The perceptron

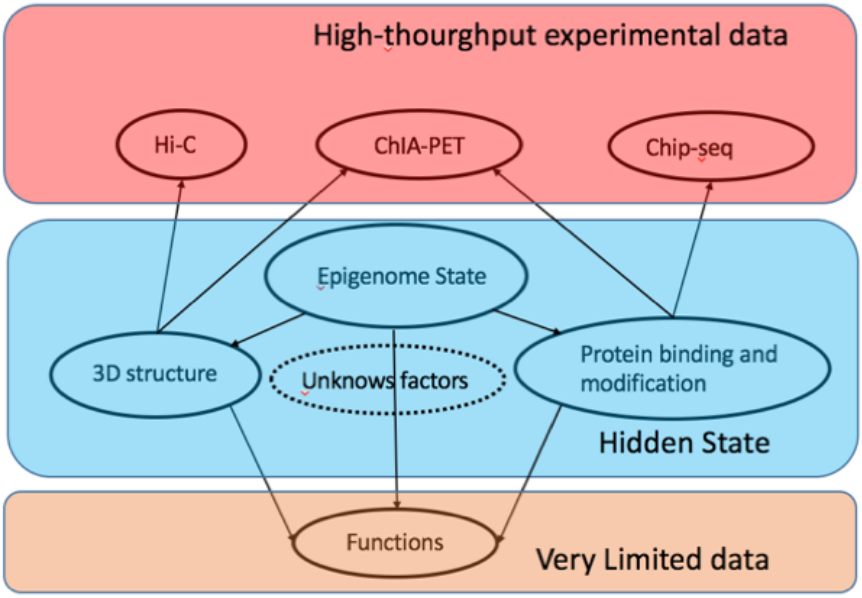


Figure 1.5: Available data in epigenomic research and corresponding relationship to the biological meanings. Hi-C, ChIA-PET and Chip-seq are major technique to detect the global states of the genome.

algorithm was invented in 1958, which enables the training of a single neuron (Rosenblatt 1958). In 1980s, Rumelhart and co-workers developed the back-propagation algorithm, which allows us to train a neural network with one or two hidden layers (D. Williams and G. Hinton 1986), and the back-propagation is still the major training strategy in today's deep learning. Before 2000s, the deep learning usually called artificial neural network (ANN). The term *deep learning* was introduced in mid-2000s when researchers can train the deeper neural network which is impossible to train before (G. E. Hinton, Osindero, and Teh 2006; Bengio, LeCun, et al. 2007; Poultney, Chopra, Cun, et al. 2007). Training a deeper neural network is the key point of this wave of the rapidly development in achine learning and artificial intelligence.

The performance of the neural network gradually increases in the past several decades. The explosive development of deep learning comes from the dramatic moment when a convolutional neural network win the Large Scale Visual Recognition

Challenge (ILSVRC) in 2012 and cut the error rate nearly in half (from 26.1% to 15.3%) (Krizhevsky, Sutskever, and G. E. Hinton 2012). In computer vision, deep learning also achieve tremendous success in object detection and image segmentation (Clement Farabet et al. 2013; Clément Farabet et al. 2012; Sermanet et al. 2013). Introducing deep learning to speech recognition also result the sudden drop in the error rate at the beginning of 2010s (Dahl et al. 2012; A.-r. Mohamed, Dahl, and G. Hinton 2012; Graves, A.-r. Mohamed, and G. Hinton 2013)

1.2.2 Deep Learning in Genomics and Epigenomics

In 2015, two individual groups (Alipanahi et al. 2015; Jian Zhou and Troyanskaya 2015) published their works to predict the locations of protein binding sites on the DNA string using deep learning from the DNA sequence only. The works open the era of deep learning in epigenomics. Finding functional related DNA patterns (DNA motif) is relied on the manually feature engineering in the past, and the deep learning technique provides a fully automatic procedure to detect such patterns. In these ground-breaking works, most of the patterns extracted by the previous work in a manual way are also discovered by the deep neural network (Alipanahi et al. 2015). Also, several new patterns are revealed to be important to the epigenomic functions. Several followed-up work using improved network topology to enhance the performance of the deep neural network (Kelley, Snoek, and Rinn 2016; Quang and X. Xie 2016; H. Zeng et al. 2016).

1.2.3 Deep learning in studying genomic 3D structures

High-throughput epigenomic data sets, including Hi-C, are usually huge in volume, so the data cannot be processed in a manual way like the traditional experimental data in the biological studies, and automatically processing pipeline is highly needed.

In addition, epigenomic data sets, including Hi-C, are usually in high dimensions,

high noise level and complicated in the structures. Also, comparing with images and natural languages, epigenomic datasets are brand new to human, making it harder to manually exploring the features in the epigenomic data.

In deep learning, the deep neural network can automatically extract patterns based on the training data sets without the need of manual intervention, making it possible to develop automated pipelines for the epigenomic data analysis. Furthermore, deep learning requires a large volume of training data, which also agree with the properties of epigenomic data sets.

Although the data amount is large, epigenomic data sets are usually not well-formatted. In deep learning (deep learning only refer to the discriminative model in this dissertation), the neural networks are employed to reflect the generalized mapping relationship between the input data X and the output data (also called labels) Y. In epigenomics, to develop a deep learning algorithm, the essential part is to properly define the X and Y based on the domain knowledge. Designing a loss function for the training that accurately reflecting the biological facts is the essential step to develop such an algorithm.

In our work, we have developed multiple algorithms to solve the fundamental problems in 3D genomics, including constructing genomic spatial structure, experimental data enhancement, and the comparative analysis of the 3D features. To our best knowledge, we are the pioneer to apply the deep learning technique to Hi-C data and 3D genomics. We focused on the demands from the biological research as well as the latest advances in the deep learning, and solve the biological problems from the basic scientific fact without many detailed assumptions. Our algorithms emphasize employing the automatically analyze pipeline to discover the biological finding with least manual interventions.

Our research outcomes have been presented in multiple academic conferences as well as preprint servers, and attracted wide interest. Several publications in academic

journals and/or academic proceedings have been accepted and under reviewing.

Chapter 2

A Flexible Approach to Reconstruct the Genomic Spatial Structure by the Genetic Algorithm

2.1 Introduction

The Hi-C technique (Lieberman-Aiden et al. 2009) is designed to detect the 3D structures of the genomes, so recovering the 3D structure from the Hi-C data is one of the fundamental task. As the development of the Hi-C technique, a lot of approaches have been proposed to reconstruct the spatial structure of chromosomes (Duan et al. 2010; Baù and Marti-Renom 2011; Tanizawa et al. 2010; Hu et al. 2013; Varoquaux et al. 2014; Rousseau et al. 2011; S. Wang, J. Xu, and J. Zeng 2015; Z. Zhang et al. 2013; Trieu and Cheng 2014; Trieu and Cheng 2015; Peng et al. 2013; Shavit, Hamey, and Lio 2014; Nowotny et al. 2015; Lesne et al. 2014). All of these reconstruction processes consist of two stages: 1) proposing an objective function to evaluate how the 3D structure reflects the Hi-C experimental data, and converting the problem to an optimization problem; 2) developing the algorithm to solve the proposed optimization problem and obtaining the 3D structures.

The objective function reflects the quantitative model for the relationship between the 3D structure and the Hi-C experimental data. Currently, several quantitative models have been proposed to reconstruct the 3D structures. However, whether these models are appropriate in all the conditions is still a controversy. Also, the Hi-

C field is rapidly developing, and new models are proposed periodically with the new discoveries. However, in most of the current work, the optimization algorithms have already integrated with the objective function. The optimization tool can only solve a particular objective function. Thus, a flexible and model-free approach is sometimes desirable. In this work, we develop a model-free optimization tool for obtaining the optimized chromosomal 3D structure using a genetic algorithm (GA).

The genetic algorithm (GA) (Davis 1991; Weise 2009) is a randomized heuristic search strategy which is inspired by the evolutionary process in nature. The genetic algorithm is widely used in the non-linear optimization problems. The objective function is named fitness function. Genetic algorithms have already successfully implemented to reconstruct 3D structures from pair-wise interaction matrix obtaining from experiments for atomic cluster (Deaven and K. Ho 1995; Chua et al. 2010) and protein (Bowie and Eisenberg 1994; Bayley et al. 1998; Ono et al. 2002; Gardiner, Willett, and Artymiuk 2001). However, since the Hi-C is a recently developed technique, very few researchers have employed the GA to solve the 3D structure of chromosomes (Nowotny et al. 2015). Comparing with other optimization strategies, the genetic algorithm has several advantages: 1) Fewer assumptions and less manual intrusions are required for the fitness function in genetic algorithm; 2) Genetic algorithm has better performance to overcome the local optimal problems than other approaches (Meza et al. 1996); 3) The optimization goal is flexible via modifying the fitness function, which is essential for a rapidly developing field where new models are raised up very often. Also, the building-block hypothesis, an important assumption of GA (Golberg 1989), is exactly fit for the current understanding of 3D genomic structures since the Hi-C experiment has revealed the local compact substructure at different scale levels (Lieberman-Aiden et al. 2009; J. R. Dixon et al. 2012; Rao et al. 2014).

2.2 Methods

In the genetic algorithm, a certain number of candidate solutions are maintained during the entire process of searching for the optimal solution. The candidate solutions' pool is named population. A single candidate solution in the pool is named an individual. The population keeps updating to obtain the optimal candidate.

The steps of genetic algorithm include: 1) Initialization, 2) Selection, 3)Mutation and Crossover, 4) Termination. Step 2 and 3 run iteratively until the termination condition is reached.

New individuals of the next generation are created by the genetic operators including crossover and mutation, in which processes, the fitter individuals have more chance to be selected while the diversity is kept. The fitness function, which reflects the optimization goals, is used to evaluate the fit level for individuals.

2.2.1 Fitness Function

In our specific problem, the fitness function is to evaluate fitness level the proposed 3D structure with the Hi-C experimental data quantitatively. In previous work, the fitness level is measured based on the total absolute error value or the probability. In the total error value model, Hi-C interaction numbers are directly converted to the spatial distances. The Hi-C interaction map is converted to the pair-wise distance matrix. Then the fitness level of proposed 3D structure is evaluated by calculating the sum of the error value between all pair-wise distances in the 3D structure and the Eq.2.1 is one of the most commonly used functions in the distance-based model (Duan et al. 2010; Baù and Marti-Renom 2011; Tanizawa et al. 2010; Hu et al. 2013; Varoquaux et al. 2014; Rousseau et al. 2011; S. Wang, J. Xu, and J. Zeng 2015).

$$Fitness\ Score = \sum_{i=1}^{n} \sum_{j=i+1}^{n} (DC_{\mathrm{ij}} - DH_{\mathrm{ij}})^2 \tag{2.1}$$

where DH_{ij} is the distance converted from the Hi-C interaction matrix and DC_{ij} is the distance computed from the proposal 3D structure.

The error value model is easy to be understood, and it is relatively less computation intensive. However, the error value model are based on the assumption that the chromosome has a single structure. However, the Hi-C experiment employs a large group of the cells. Alternative methods for the 3D chromosome structure applied the statistical approaches to building probabilistic model (Hu et al. 2013; Varoquaux et al. 2014; Rousseau et al. 2011; S. Wang, J. Xu, and J. Zeng 2015). Eq. 2.2 is an example of using Poisson distribution to obtain the probability of the proposed structures. The Hi-C experiment utilized a large group of cells. The number of interactions observed for each pair of location is following binomial distribution. In the Hi-C experiment condition, the number of cells is enormous, and the probability of ligation is small. The binomial distribution is approximate to Poisson distribution. Therefore, Poisson probability model is the better fit to the nature of the Hi-C experiment.

$$Fitness\ Score = -\sum_{i=1}^{n}\sum_{j=i+1}^{n} log(P(D_{\mathrm{ij}}, E_{\mathrm{ij}})) \tag{2.2}$$

where P is the Probability Density Function(PDF) of Poisson distribution, D_{ij} is the actual Hi-C interaction number (integer) and E_{ij} is the expected Hi-C interaction number from proposal structure (float).

In both of the approaches, the Hi-C interaction numbers need to be converted to distances or expected distances. In our GA framework, the conversion equation is determined by the user. For testing purpose, we use Eq. 2.3 the make the conversion.

$$DH_{\mathrm{ij}} = \frac{1}{{H_{\mathrm{ij}}}^{\mathrm{n}} + const}, \tag{2.3}$$

where DH_{ij} is the distance converted from the Hi-C interaction matrix, H_{ij} is the Hi-C interaction number , and *const* is parameter to control the conversion

2.2.2 Detailed Algorithm

The initial population is generated by the random process. If no 3D structure from the other sources is imported, all of the initial structures at the beginning of the genetic algorithm are generated by random number generator within a certain scale to spread the population across the large searching space. If needed, the genetic algorithm can also adopt the result from other optimization tools to perform the further optimization. In this case, the initial population composed both of the random generators as well as the 3D structure from user input.

Selection Strategy

The principle of the selection is that better individuals have larger chance to mate and pass their properties to the next generation. At the beginning of the selection process, all of the current candidates are sorted based on the fitness scores. The best candidates (usually around top 1%) in the candidate's pool are directly selected to add the next generation's pool without any change. Then a roulette wheel process is performed to choose the candidates to the crossover and mutation process. The selection is based on the ranking of the candidate solution, and a ranking score is assigned as Eq. 2.4.

$$Ranking\ Score(RS) = \frac{1}{rank + constant\ number} \tag{2.4}$$

where *rank* is the position of the candidate sorted by fitness score, *constant* is the parameter to adjust the generation gap.

For each selection, the probability of an individual to be selected is proportional to its rank score and probability of selection is shown in Eq. 2.5

$$P_{\mathrm{i}} = \frac{RS_{\mathrm{i}}}{\sum\limits_{j=N}^{1} RS_{\mathrm{j}}} \tag{2.5}$$

where P_{i} is the probability of individual i to be selected, RS_{i} is ranking score of individual i, and N is size of the total candidate pool.

Each selection process is independent and doesn't change the status of the individual pools. The selection process keeps running to pick the candidates solution as parents for the crossover until crossover stage finishes.

Crossover

Crossover is analogous to reproduction in the biological crossover. Children are generated based on its parents in the crossover. In our work, two candidate solutions are picked up independently from the selection process and two children are produced from a pair of parents by the crossover. Assume two parents are P_1 and P_2, two children are C_1 and C_2, and the number of the beads for each is k. The $P[i]$ indicates the spatial position of ith bead on the candidate solution P. Steps of the crossover includes: 1) random numbers R between 2 to $K-1$ is generated; 2) Translocate the spatial position of P_1 and P_2 to make $P_1[R]$ and $P_2[R]$ are both $(0,0,0)$; 3) Produce child C_1 with $P_1[1...R]$ and $P_2[R+1...K]$ and produce child C_2 with $P_2[1...R]$ and $P_1[R+1,K]$. 4) The fitness score is calculated for C_1 and C_2 based on their new 3D coordinates.

Mutation

The mutation is performed in place on the children from the crossover. In the mutation operation, the particular percentage of 3D points are aligned to the new location based on the normal distribution with the mean value being their original position. The percentage of individuals selected, the proportion of beads changed and the deviation of the distribution are determined by the user, and they are relevant parameters in a genetic algorithm.

2.3 Experimental Results

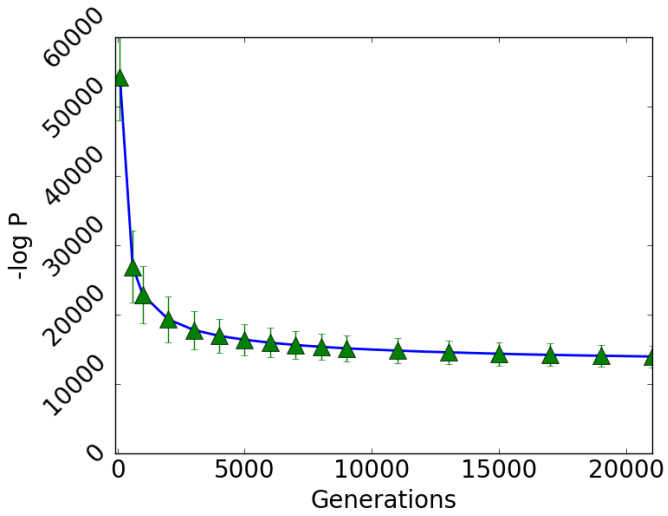


Figure 2.1: Performance of the independent experiments under the identical condition. We employed the same Hi-C interaction data, the same model and the same parameters to run independent experiments for 500 times to investigate the stability of our implementation of the genetic algorithm between different runs.

2.3.1 Consistency of genetic algorithm

Similar to the other heuristic methods, the genetic algorithm involves a random process in several steps. Therefore, different runs for the same program with identical parameters and inputs may lead to different results. It is important to study the difference between same sample and parameters in multiple runs. The difference between independent experiment with the same input and same setting should be within a reasonable range. If multiple parallel runs give a quite different result, the possible reason is the result not reached the converge points or the result was trapped at local optimal. The parameter needs to be improved.

To test the stability of performance from run to run, we ran the genetic algorithm 500 independent experiments at the exact same condition, including the inputs, mathematical model and parameters. We employed the probability model based on Poisson distribution. The fitness score is calculated by the minus log conditional probability

in Eq. 2.2 The result is shown in Fig. 2.1. The curve is the average of all of the runs at each point, and the bar is the two times of the standard deviation. As expected, at the begining, the result of each run is quite different with each other because of the random initialization. As the selection pressure from the objective function and selection process, the results tend to be similar. The result reach converges at around 10000 generations since the deviation and average are not decreasing anymore.

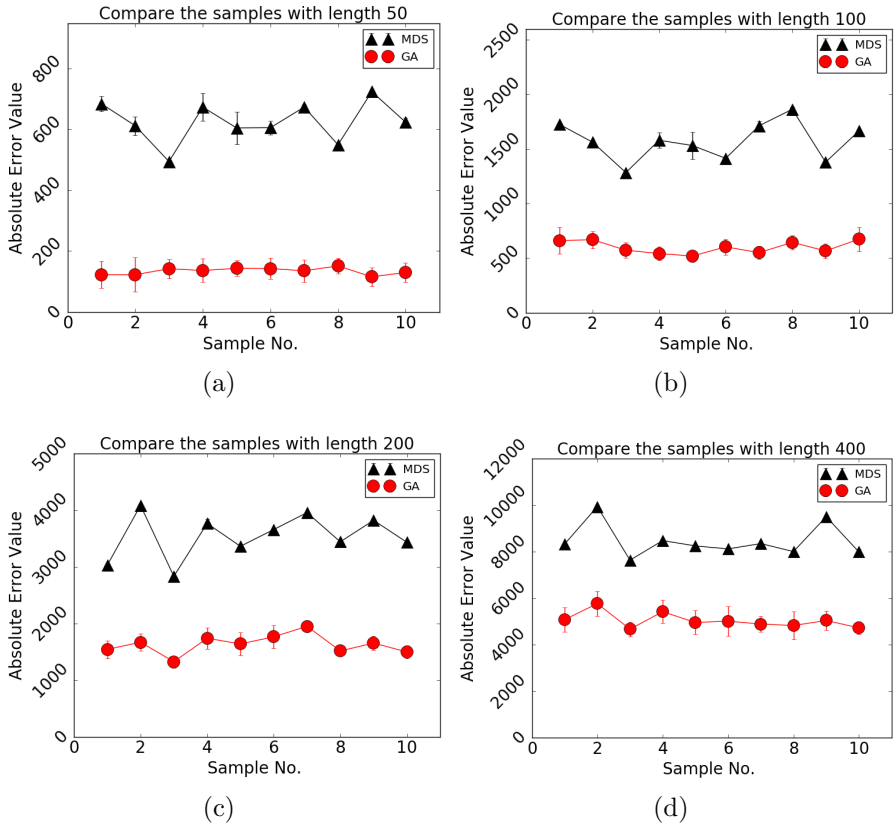


Figure 2.2: Performance of the MDS method and the Genetic Algorithm (GA) for minimizing the error value for different size of problems.

2.3.2 Comparing with MDS-based method

Several approaches convert the 3D structure recovery problem to a multidimensional-Scaling (MDS) problem and solve MDS problems (Lesne et al. 2014; Segal and Bengtsson 2015; Varoquaux et al. 2014). Some mathematically software packages, such

as C++ GNU IPOPT (Wächter and Biegler 2006) and Python Scikit (Pedregosa et al. 2011), can be directly utilized to solve the error minimization problem. To compare the performance between MDS-base approach and the genetic algorithm head-to-head, we implement the same goal function (Eq.2.1) for both optimization approaches. We studied the performance of the MDS and the GA at four different scales. For each scale, we randomly chose 10 qualified samples from the entire Hi-C interaction map. For each sample, both GA and MDS experiments were carried out ten times to get the statistical result as shown in Fig. 2.2. The MDS package is from the Python Scikit's method (Pedregosa et al. 2011) and all of the parameters is default except the number of steps. The parameters for the genetic algorithm are also default without particular optimization. The bar is two times of standard deviation. At all of the cases, the genetic algorithm outperforms the conventional MDS-based method. It is worth to notice that the standard deviation is nearly zero for many MDS results, indicating the MDS optimization method quickly falls into the same local optimal.

2.3.3 Comparing with the other approaches

We also compared the performance of our approach with the other approaches in the similar model. Hu and coworkers (Hu et al. 2013) developed Bayesian 3D constructor for Hi-C data (BACH) to infer the consensus 3D chromosomal structure. The object function of the BACH model is based on the Poisson probability density function and also incorporate the physical properties of the genome to improve the model accuracy. The model also includes a series of nuisance parameters which are also considered in the optimization step. The objective function is shown in Eq. 2.6 and 2.7.

$$logP = \sum_{1<i<j<n} PDF_{\text{Poisson}}(U_{\text{ij}}, \theta_{\text{ij}}) \tag{2.6}$$

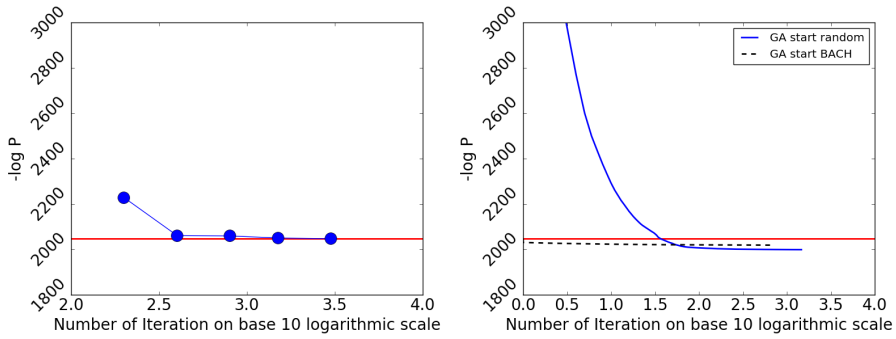
where P is probability of the 3D structure at given condition, and PDF is probability density function

$$log\theta_{\mathrm{ij}} = log(\theta_0 + \theta_1 log(d_{\mathrm{ij}}) + \theta_2 log(e_{\mathrm{i}} e_{\mathrm{j}}) + \theta_3 log(g_{\mathrm{i}} e_{\mathrm{j}}) + \theta_4 log(m_{\mathrm{i}} e_{\mathrm{j}})) \tag{2.7}$$

where is u_{ij} is the Hi-C interaction number between loci i and j, θ_{i} is Nuisance parameter , and the$_{\mathrm{i}}$, g_{i}, m_{i} are properties of the chromosome and refer to (Hu et al. 2013) for detail.

The BACH employs Markov chain Monte Carlo (MCMC) to maximize the probability and obtain the 3D structure. The process optimizes 3D structure and nuisance parameters iteratively. To compare the performance between the original MCMC method and the genetic algorithm, we implemented the same objective function as the fitness function in the genetic algorithm. Also, we configured our program to optimize the 3D structures and nuisance parameters by the genetic algorithm iteratively. The result is shown in Fig. 2.3. The samples input provided by Hu (Hu et al. 2013) was employed as the input for both BACH and GA test. We first ran BACH for different iteration to observe the results. The BACH uses an efficient MCMC method to obtain the 3D structure and nuisance parameters. Fig. 2.3a shows the result convergence very quick at the early stage of the process and the computation speed is very fast. However, the BACH approaches stuck in local optimal and cannot further improve. We start from the random initialization and show the result in the Fig. 2.3b. The genetic algorithm initially converges slowly but eventually the result outperform the BACH approach. To facility the genetic algorithm, we implemented an improved version of the genetic algorithm to take advantage existing BACH result. As shown in Fig.Hi-C 2.3b, both accuracy and converge speed are improved if we take BACH result as starting point of the genetic algorithm.

We also compared the performance in accuracy with PASTIS (Varoquaux et al. 2014), which also applied the Poisson model as the objective function. We evaluated



(a) (b)

Figure 2.3: Compare the performance of GA and BACH approaches based on the same model in original BACH algorithm. For comparison purpose, the horizontal bar is the best result obtained by BACH

the probability of the structure calculated by the PASTIS package in Fig. 2.4. The PASTIS converge speed is very quick, so we just draw a horizontal bar to represent the optimal result from the PASTIS. When the computation time is short, the PASTIS generates better result comparing with the genetic algorithm. After computing several thousands of iteration, the genetic algorithm eventually outperforms the PASTIS in accuracy.

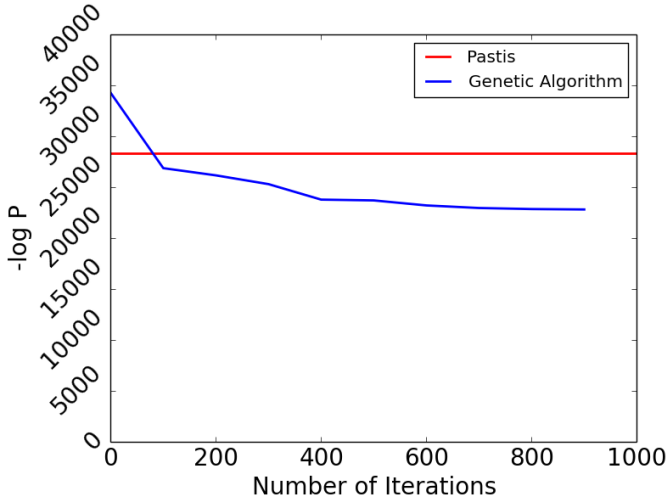


Figure 2.4: Compare the performance of GA and PASTIS. The red line is the performance provided by PASTIS, and the blue line is the performance of our GA approach. The performance of GA exceed that of PASTIS after 100 iterations

2.4 Discussion

Our approach employs the genetic algorithm to reconstruct the spatial structure from the Hi-C interaction data. Our approach outperforms the performance of several of the current approaches in accuracy under the same biological model. Besides the advantage in accuracy, our approach is flexible to all kinds of the biological model, which is of great importance in such a rapidly developing field. It is easy to switch the biology model between distance model and probability model. The relevant software package (GA3D) can run stand alone as well as work together with results from other software to facility the computation speed and obtain the better result.

Chapter 3

HiCPlus: a deep convolutional neural network for Hi-C interaction matrix enhancement

3.1 Introduction

The Hi-C technique (Lieberman-Aiden et al. 2009) has emerged as a powerful tool for studying the spatial organization of chromosomes, as it measures all pair-wise interaction frequencies across the entire genome. In the past several years, Hi-C technique has facilitated several exciting discoveries, such as A/B compartment (Lieberman-Aiden et al. 2009), Topological Associating Domains (TADs) (J. R. Dixon et al. 2012; Nora et al. 2012), chromatin loops (Rao et al. 2014) and Frequently Interacting Regions (FIREs) (A. D. Schmitt et al. 2016), and therefore significantly expanded our understanding of 3D genome organization (Lieberman-Aiden et al. 2009; J. R. Dixon et al. 2012; Rao et al. 2014) and gene regulation machinery (A. D. Schmitt, Hu, and Ren 2016).

Hi-C data is usually presented as a $n \times n$ contact matrix, where the genome is divided into n equally sized bins and the value within each cell of the matrix indicates the number of pair-ended reads spanning between a pair of bins. Depending on sequencing depths, the commonly used sizes of these bins can range from 1 kb to 1 Mb. The bin size of Hi-C interaction matrix is also referred to as '*resolution*', which is one of the most important parameters for Hi-C data analysis, as it directly affects the results of downstream analysis, such as predicting enhancer-promoter interactions or identifying TADs boundaries.

Sequencing depth is the most crucial factor in determining the resolution of Hi-C data - the higher the depth, the higher the resolution (thus the smaller bin size). Due to high sequencing cost, most available Hi-C datasets have relatively low resolution such as 25 or 40 kb, as the linear increase of resolution requires a quadratic increase in the total number of sequencing reads (A. D. Schmitt, Hu, and Ren 2016). These low resolution Hi-C datasets can be used to define large-scale genomic patterns such as A/B compartment or TADs, but cannot be used to identify more refined structures such as sub-domains or enhancer-promoter interactions. Therefore, it is urgent to develop a computational approach to take full advantage of these currently available Hi-C datasets to generate higher resolution Hi-C interaction matrix.

Recently, deep learning has achieved great success in several disciplines (LeCun, Bengio, and G. Hinton 2015; Schmidhuber 2015; Goodfellow, Bengio, and Courville 2016), as well as computational epigenomics (Koh, Pierson, and Kundaje 2017; Angermueller et al. 2016; Schreiber et al. 2017; F. Liu et al. 2016). In particular, Deep Convolutional Neural Network (ConvNet) (LeCun, Bengio, and G. Hinton 2015; LeCun et al. 1998), which is inspired by the organization of the animal visual cortex (LeCun et al. 1998; Fukushima 1980; Serre et al. 2007), has made major advancement in computer vision and natural language processing (LeCun et al. 1998). In the fields of computational biology and genomics, ConvNet have been successfully implemented to predict the functional targets of DNA sequence (Jian Zhou and Troyanskaya 2015; Alipanahi et al. 2015; Kelley, Snoek, and Rinn 2016; H. Zeng et al. 2016; Quang and X. Xie 2016; Jiyun Zhou et al. 2016) as well as the epigenetic states (e.g. methylations and gene expression levels) from experimental assays (Singh et al. 2016; Angermueller et al. 2017; Min et al. 2016; Y.-z. Zhang et al. 2017).

In this chapter, we propose HiCPlus, which is the first approach to infer high-resolution Hi-C interaction matrices from low-resolution or insufficiently sequenced Hi-C samples. Our approach is inspired by the most recent advancements (Glasner,

Bagon, and Irani 2009; J. Yang et al. 2008; Dong et al. 2016; Dong et al. 2014) in the single image super-resolution, and can generate the Hi-C interaction matrices with the similar quality as the original ones, while using as few as 1/16 of sequencing reads. We observe Hi-C matrices are composed by a series of low-level local patterns, which are shared across all cell types. We systematically applied HiCPlus to generate high-resolution matrix for 20 tissue/cell lines (total 22 replicates) where only low resolution Hi-C datasets are available, covering a large variety of human tissues. In summary, this work provides a great resource for the study of chromatin interactions, establishes a framework to predict high-resolution Hi-C matrix with a fraction of sequencing cost, and identifies potential features underlying the formation of 3D chromatin interactions.

3.2 Method

3.2.1 Overview of the framework

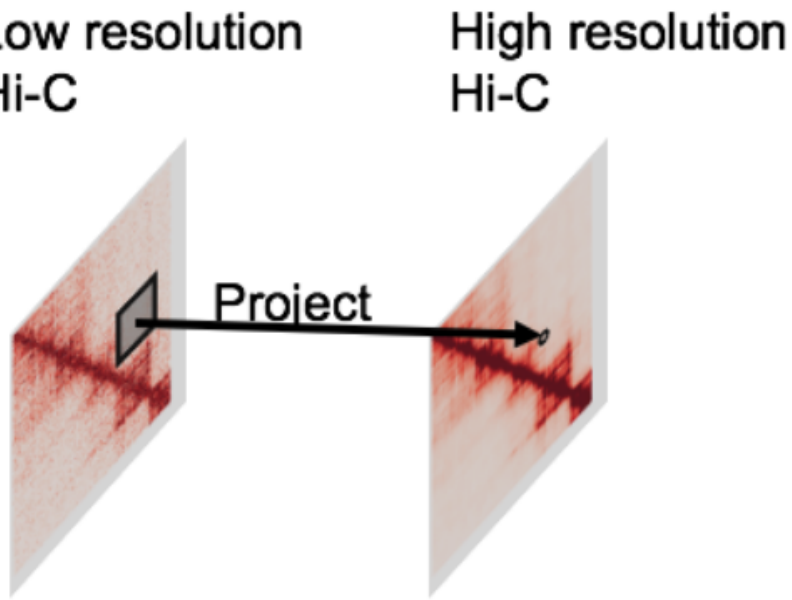


Figure 3.1: HiCPlus leverages information from surrounding regions to estimate contact frequency for a given point in a Hi-C interaction matrix.

To obtain the insufficiently sequenced Hi-C samples, we down-sample the sequencing reads to 1/16 and construct another interaction matrix at the same resolution ,which consequently contains more noises and more blurred patterns. We then fit the ConvNet model using values at each position in the high-resolution matrix as the response variable and using its neighboring points from the down-sampled matrix as the predictors (Fig. 3.1). Our goal is to investigate whether the ConvNet framework can accurately predict values in the high-resolution matrix using values from the low-resolution matrix. Noticeably, although technically both matrices are at the same resolution, we consider the down-sampled interaction matrix "*low-resolution*", as in practice, it is usually processed at lower resolution due to the shallower sequencing depths. In this paper, we use"*low resolution*" and "*insufficiently sequenced*" interchangeably.

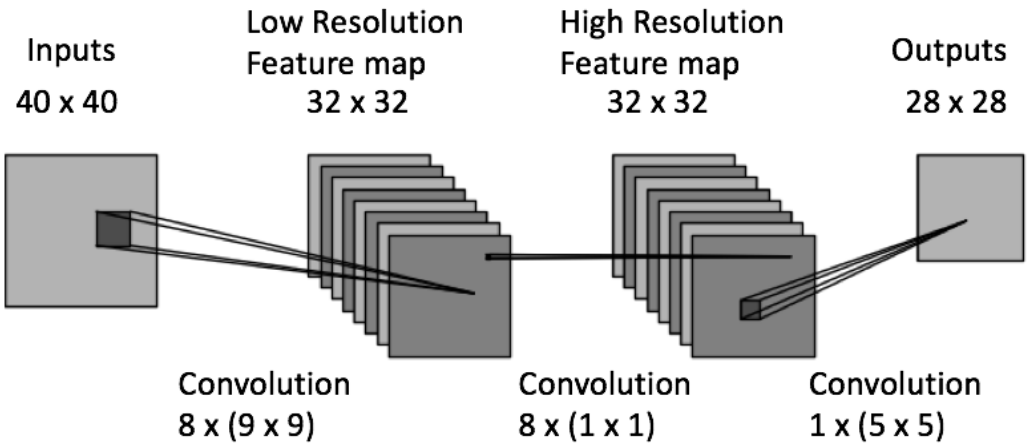


Figure 3.2: The topology of the convolutional neural network in HiCPlus HiCPlus contains three convolutional layers, and the output of the third convolutional layer is the output of overall neural network. The hyper-parameters listed here are used throughout this work unless otherwise noted.

Detail Method

1. Pre-processing Hi-C matrix: Many of the current available Hi-C data, especially in human tissue, are only available at 40kb resolution matrices. For these

data sets, we start from the BAM file and generate the10Kb resolution interaction matrices. Consequently, we observe an increased noise-to-signal ratio comparing with deeply sequenced Hi-C library. In the training stage, we start from high-resolution Hi-C data (such as GM12878 from `https://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE63525`) and use a random down-sampling method to simulate the low-resolution Hi-C matrix. After this step, all input matrices are at 10Kb resolution. As previously mentioned, we consider the matrices generated from down-sampled sequencing reads as low-resolution since they would have been processed at a lower-resolution at that sequencing depths in practice. .

2. Divide a Hi-C matrix into multiple square-like sub-regions with fixed size, and each sub-region is treated as one sample. Unless otherwise noticed, each sub-region is $0.4Mb \times 0.4Mb$ which contains $40 \times 40 = 1,600$ pixels at 10Kb resolution. We only investigate and predict chromatin interactions where the genomic distance between two loci is less than 2Mb, as the average size of TADs is less than 1Mb and there are few meaningful interactions outside TADs.

3. The deep ConvNet is trained to learn the relationship between the low-resolution samples (a.k.a same size but insufficient sequenced samples) and high-resolution samples in the training stage, and predicts the high-resolution samples from low-resolution samples in the production stage.

4. The predicted high-resolution sub-matrices are merged into chromosome size Hi-C interaction matrix. As the samples have a surrounding padding region that is removed during the prediction by ConvNet, the proper overlap is necessary when dividing the Hi-C interaction matrix to the samples in the Step 1

For the ConvNet, the input is a list of low-resolution samples with $N \times N$ size for each sample. To avoid the border effect, similar with Dong's work(Dong et al. 2016), we didn't add white padding to any convolutional layer so the output of each sample has the smaller size. Therefore, the output is a list of predicted high-resolution samples with $(N - padding) \times (N - padding)$ size, where $N = 40$ and $padding = 12$ for the typical setting in this discussion, and both input 40×40 matrix and output 28×28 matrix are registered in the same central location. The shrunk size can be offset by the overlapping during the dividing process. We denote the ConvNet model as F, the low-resolution input as X, the predicted high-resolution output as Y, and the real high-resolution Hi-C as Y (Y is also regarded as ground truth in this section). Mean Square Error (MSE) is used as loss function (Eq. 3.1) in the training process. Therefore, the goal of the training process is to generate F that minimizes the MSE.

$$argmin\frac{1}{m}\sum_{i=1}^{m}(F(X_i) - Y)^2 \tag{3.1}$$

As shown in Fig 3.3, the ConvNet in HiCPlus has three layers, serving for extracting and representing patterns on the low-resolution matrix, non-linearly mapping the patterns on the low-resolution matrix to high-resolution matrix, and combining the high-resolution patterns to generate the predicted matrix, respectively. Below we describe each layer in detail.

Pattern extraction and representation In this stage, input is the low-resolution $f_1 \times f_1$ matrix, and output is generated by the following Eq. 3.2

$$F_1(X) = max(0, w_1 * X + b_1) \tag{3.2}$$

where $*$ denotes the convolutional operation, X is the input matrix, b_1 is the bias, and w_1 is an $n_1 \times f_1 \times f_1$ matrix. Here n_1 and f_1 are the filter numbers and filter size, respectively. Both n_1 and f_1 are hyper-parameters in the ConvNet, and we set

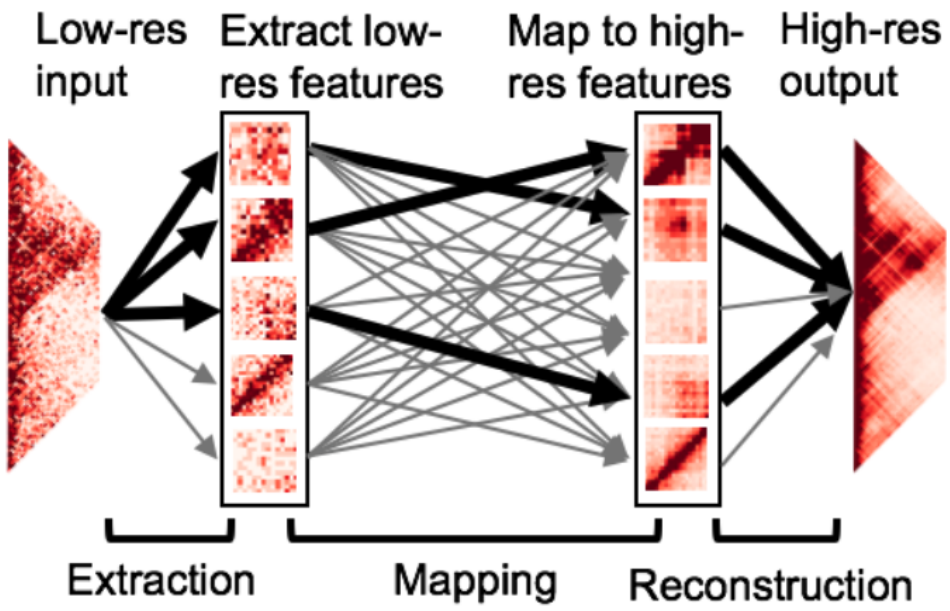


Figure 3.3: Conceptual view of the network structure in HiCPlus: regional interaction features (e.g. loops, domain borders) are learned using values at each position in the high-resolution matrix as the response variable and using its neighboring points from the low-resolution matrix as the predictors.

n_1 to 16 and f_1 to 5. As shown in (Fig. 3.2c), HiCPlus is not sensitive to these hyper-parameters. The Rectified Linear Unit (ReLU)(Nair and G. E. Hinton 2010) is utilized as the non-linear activation function.

Non-linear mapping between the patterns on high-resolution and low-resolution maps This stage is shown as the middle part of the Fig. 1(b), where the patterns on the low-resolution matrix are mapped non-linearly with the patterns on high-resolution matrix using the formula:

$$F_2(X) = max(0, w_2 * F_1(X) + b_2) \tag{3.3}$$

where $F_1(X)$ is the output from the previous layer, b_2 is the bias, and w_2 are n_2 matrices, each has the size of $f_2 \times f_2$. In this layer, we set n_2 to 16 and f_2 to 1 as it is a process of non-linear mapping.

Combining the high-resolution patterns to generate the predicted high-resolution maps We employ the following formula to generate the predicted high-resolution Hi-C matrix from the results of the second layer

$$F_3(X) = max(0, w_3 * F_2(X) + b_3) \tag{3.4}$$

where $F_2(X)$ is the output from the previous layer, b_3 is the bias, and w_3 are n_3 matrices of size $f_3 \times f_3$. In this step, the non-linear activation function is not required, and the filter number n_3 is set to 1 to generate the predicted results. Overall, function F has $parameters = w_1, w_2, w_3, b_1, b_2, b_3$. The goal of the training process is to obtain the optimal to minimize MSE on the samples in the training set. We employ the standard backpropagation(LeCun et al. 1998) with gradient descent to train the network, and use Stochastic Gradient Descent (SGD) as the update strategy. The initial parameters are drawn from the uniform distribution with Glorot?s strategy unless otherwise noted. The training is converged and no over-fitting is observed (Fig. 3.4).

3.3 Results

We describe the conceptual view of the ConvNet in Fig. 3.3, which learns the mapping relationship between high-resolution Hi-C matrix and low-resolution Hi-C matrix at feature levels. Once the model is trained, we can apply it to enhance any Hi-C interaction matrix with low-sequencing depth. HiCPlus divides the entire Hi-C matrix into small square samples and enhance them separately. After each block of interactions are predicted, those blocks are merged into chromosome-wide interaction matrix Fig. 3.5.

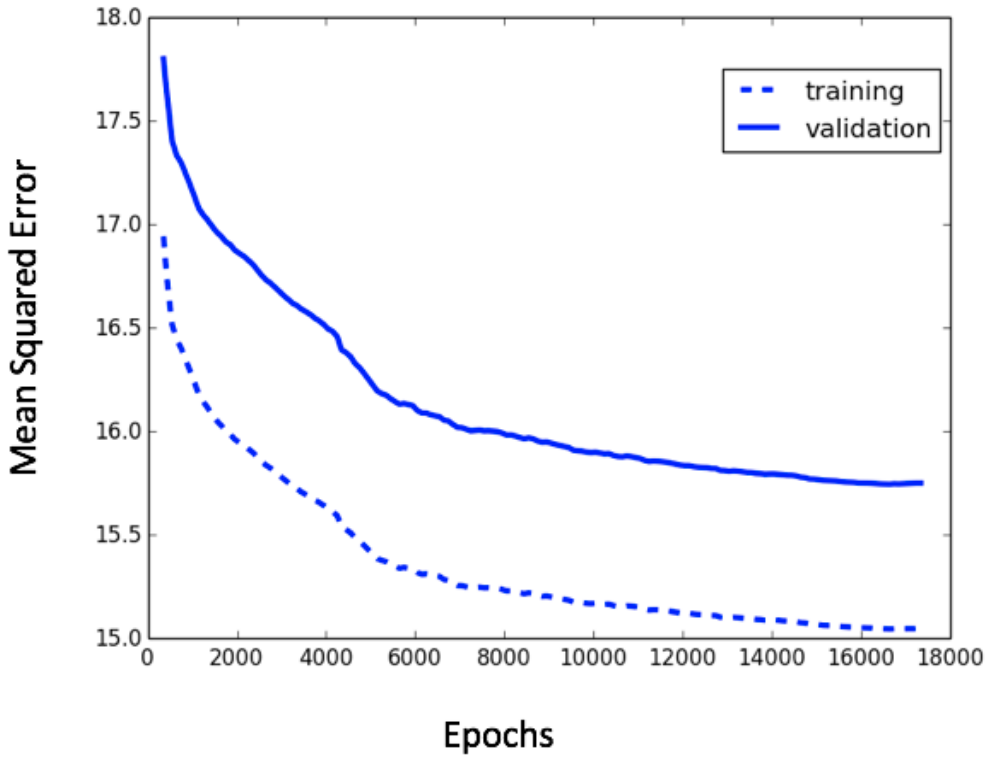


Figure 3.4: Estimation of overfitting in HiCPlus model Estimation of overfitting in HiCPlus model. To study the possible over-fitting issue in our model, we calculate the losses, which are measured in Mean Squared Error (MSE), during the training process on the training sets (chromosome 1-8) and validation sets (chromosome 19-22) in GM12878 cell line. We observe that the loss in training and training keep the same trend in the entire training process.

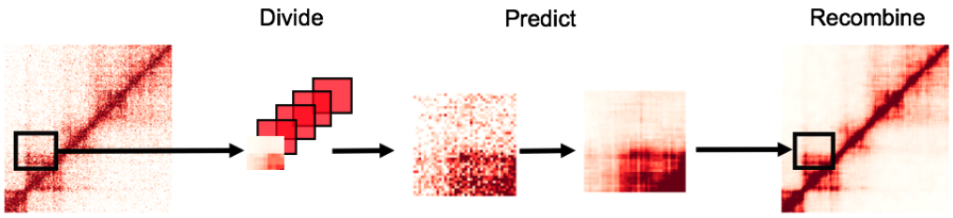


Figure 3.5: HiCPlus divides the entire Hi-C matrix into small square samples and enhance them separately. After each block of interactions are predicted, those blocks are merged into chromosome-wide interaction matrix.

3.3.1 Chromatin interactions are predictable from their neighboring regions

The hypothesis in the design of our network topology is that the Hi-C matrix contains repeating local patterns, and the interaction intensity of each point is not independent to its local neighboring regions. Base on the hypothesis, we should be able to predict the interaction frequency of any cell in the Hi-C matrix with the interaction frequencies from its neighbouring regions. To test this hypothesis, we trained a ConvNet model on chromosomes 1-17 and systematically predicted interaction matrices in chromosomes 18-22, using the 10kb resolution Hi-C data in GM12878 cells (Rao et al. 2014). To evaluate the performance of our ConvNet model, we computed both the Pearson and Spearman correlation coefficients between the predicted values and the real values at each genomic distance.

An important parameter in our model is the size of neighboring regions: intuitively, to predict the value of one point, using a larger surrounding matrix will yield better results. Therefore, we tested a range of neighboring matrix sizes, from 3×3 to 13×13 Indeed, we observed that using a larger neighboring matrix generally increases the prediction accuracy. When using a 13×13 surrounding matrix, the Pearson correlations between the predicted and real interaction frequencies are consistently higher than the predictions using smaller surrounding matrices, at each genomic distance. For example, the Pearson correlation at 40 kb genomic distance for 13×13, 7×7 and 3×3 matrices are 0.93, 0.92, and 0.89 respectively (Fig. 3.6). We tried another simple approach, by predicting each interaction frequency using the average values from its surrounding matrix. To investigate the optimal region employed for averaging, we average nearby 3×3, 5×5, 7×7 and 9×9 regions, and the averaging 3×3 could obtain the best performance (Fig. 3.7). Then, we compare the ConvNet enhanced matrix with the optimal simple averaging approach, and ConvNet performs much better than this simple approach (Fig. 3.6). In addition, we found that the prediction

accuracy reached a plateau when we used the 13×13 matrix prediction model, and further increasing the size of surrounding matrix shows little if any improvement of the prediction accuracy (Fig. 3.8). Also, we test several other potential approaches, including 2D Gaussian smoothing, Random Forest Repressor, Support Vector Repressor as shown in Fig. 3.7, and confirm the ConvNet is the optimal for prediction of the Hi-C pixel.

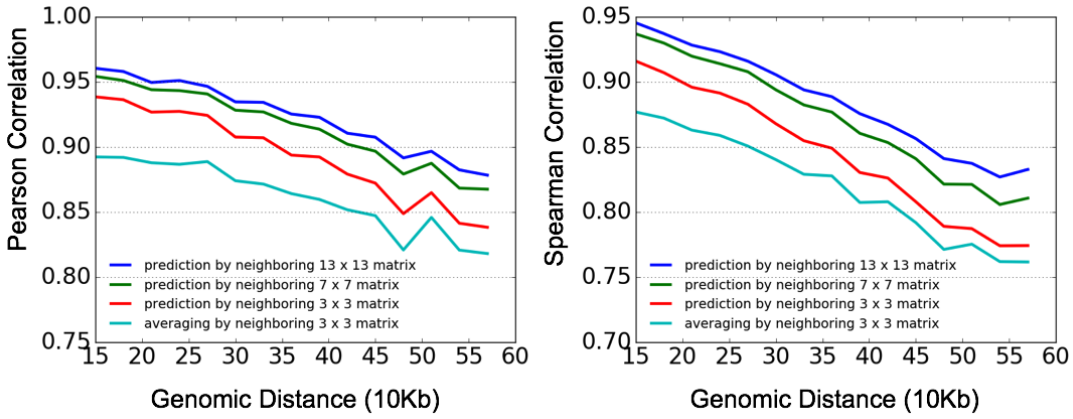


Figure 3.6: Predicting chromatin interactions from their neighboring regions We trained a ConNet model on chromosome 1-17 and systematically predicted interaction matrices in chromosome 18-22, using the 10kb resolution Hi-C data in GM12878 cell line. We used three surrounding regions sizes (3×3 , 7×7, 13×13) for prediction, and also compared their performances with a naive prediction method that simply averages the neighboring 3×3 matrix. We observe that using 13×13 matrix achieve the best performance at each genomic distance when evaluated by both Pearson and Spearman correlations.

Finally, we compared the performance of training one model for the whole matrix with training a model for each genomic distance. As it is known that there is distance decay in the Hi-C interaction matrix, which means that the further away a bins is from the diagonal of the matrix, the smaller value it tends to be. We wanted to investigate whether the distance effect has been captured in the convolutional network and whether it could affect the final output. Therefore, we trained another set of models, with each model corresponding to chromatin interactions at each distance (10kb, 20kb, and so on). We didn't observe improved performance by training models

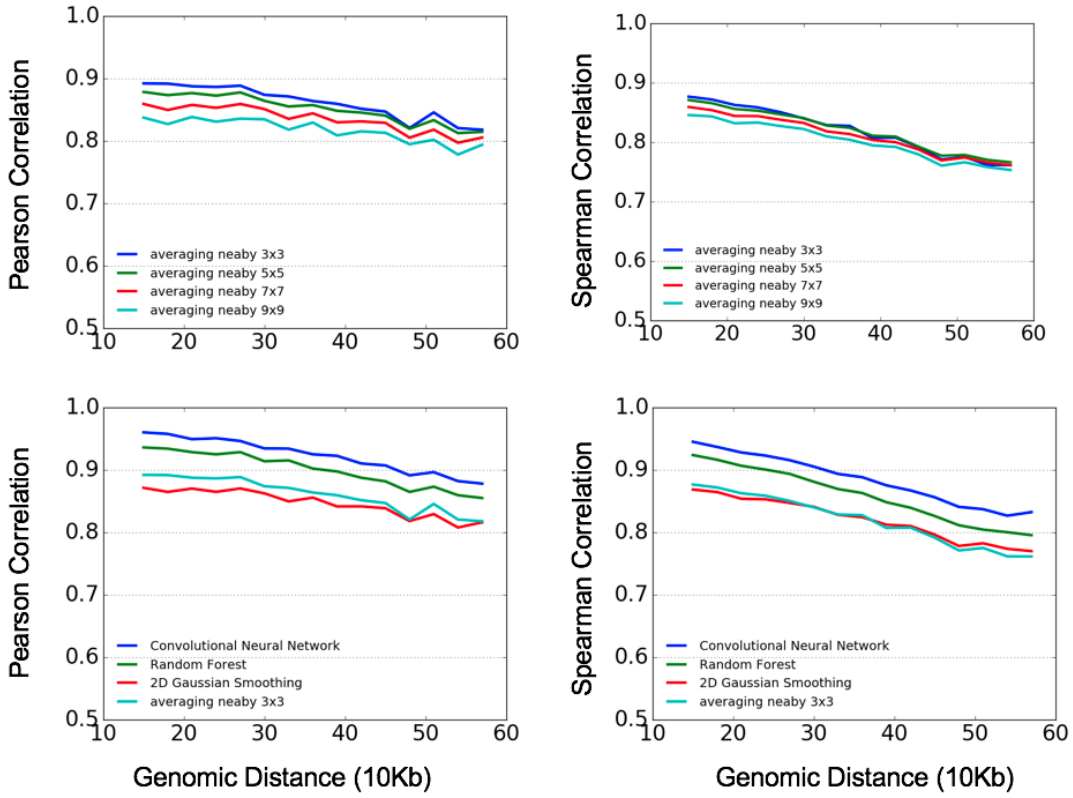


Figure 3.7: Testing the effect of using different approaches to predict chromatin interaction using neighboring regions. This figure is similar to Fig. 3.6, but we evaluate more approaches. In the upper part, to find the optimal range of averaging operation, we study different range of averaging, and the averaging 3×3 obtains best result so in Fig. 3.6, the we plot averaging 3×3 as one of the baselines. In the lower part, we add Random Forest and 2D Gaussian Smoothing to the comparison, and convolutional neural network obtains best result, inspiring us to implement HiCPlus using convolutional neural network. We also implemented Support Vector Repressor but the result is far below the curves on current plot. All of the evaluations are done in chromosome 18-22, using the 10kb resolution Hi-C data in GM12878 cells. If the model need training (e.g. Random Forest and Convolutional Neural Network), the training sets are from chromosome 1-17.

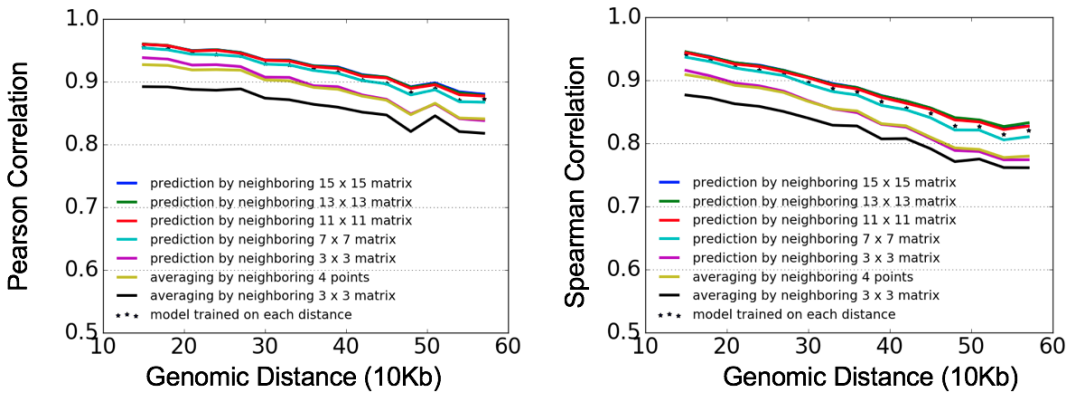


Figure 3.8: Testing the effect of using different sizes of neighboring regions to predict chromatin interaction This figure is similar to Fig. 2, but with more choices of surrounding regions for both HiCPlus and prediction by averaging nearby points. The ConvNet model is trained on chromosome 1-17 and the prediction is done in chromosome 18-22, using the 10kb resolution Hi-C data in GM12878 cells.

at different distances(Fig. 3.8), indicating that our current model has incorporated the distance effect and it is not necessary to train different models at different genomic distances. Therefore, we decided to train a single model for the whole Hi-C interaction matrix rather than training different models at different genomic.

3.3.2 Enhancing chromatin interaction matrix with low-sequence depth

Having established that values in Hi-C matrix can be predicted using their surrounding regions, we then investigated whether we can predict these values with insufficiently sequenced samples. For this purpose, we first built and tested our HiCPlus model in the same cell type, using the high resolution Hi-C data in GM12878 cell (`https://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE63525`) (Rao et al. 2014). We first constructed the 10 kb resolution matrix using all the reads (Fig. 3.9a, right panel). Then we down-sampled the reads to 1/16 of the original sequencing depth and construct the interaction matrix at the same resolution (Fig. 3.9a, left panel). The newly generated matrix contains lots of noise and TAD structures are less clear. Next, we fit a ConvNet model, using values at each bin on the high-quality

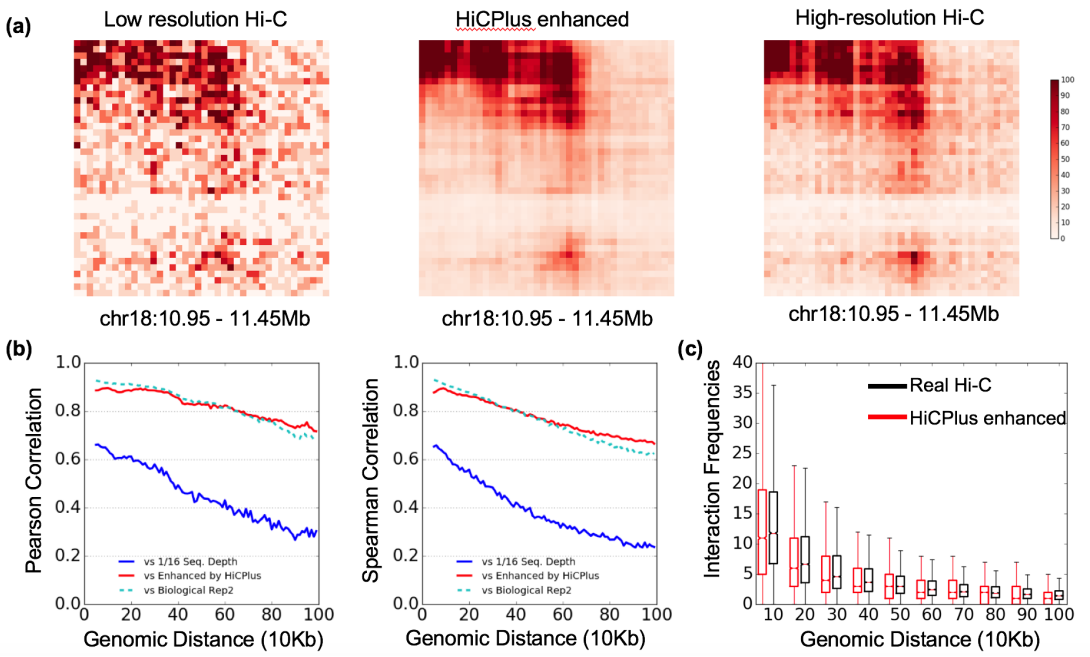


Figure 3.9: HiCPlus accurately enhances interaction matrix with low-sequence depth We trained model on chromosome 1-7 and tested the prediction in chromosome 18, in the same cell type (GM12878) at 10kb resolution. For prediction, we random chose 1/16 reads from the original total reads, built an interaction matrix (a, left panel), and then used HiCPlus to enhance it (a, mid panel). a, HiCPlus enhanced HiC and real high-resolution Hi-C matrices are highly similar. b, High correlations between HiCPlus enhanced HiC and real high-resolution Hi-C matrices each genomic distance, and they are close to the correlations between two biological replicates (dotted line). Their correlations with down-sampled Hi-C matrix is much lower (solid blue line). (c) Distribution of the Hi-C interaction frequencies at each distance for real Hi-C and HiCPlus enhanced matrices are similar.

matrix as response variable and using its neighbouring 13×13 points in the down-sampled matrix as predictors. Once the model is trained, we applied it to enhance the down-sampled interaction matrix in chromosome 18. An example of ConvNet enhanced matrix is shown in Fig. 3.9a (middle panel). We observed that the HiCPlus enhanced matrix is highly similar with the real high-resolution Hi-C matrix. Comparing with the matrix generated from down-sampled reads, it contains much less noise and both the individual chromatin loops and the TAD structures are more visible.

To quantitatively evaluate the performance of HiCPlus, we computed the Pearson correlation and Spearman ranking correlation between the experimental high-

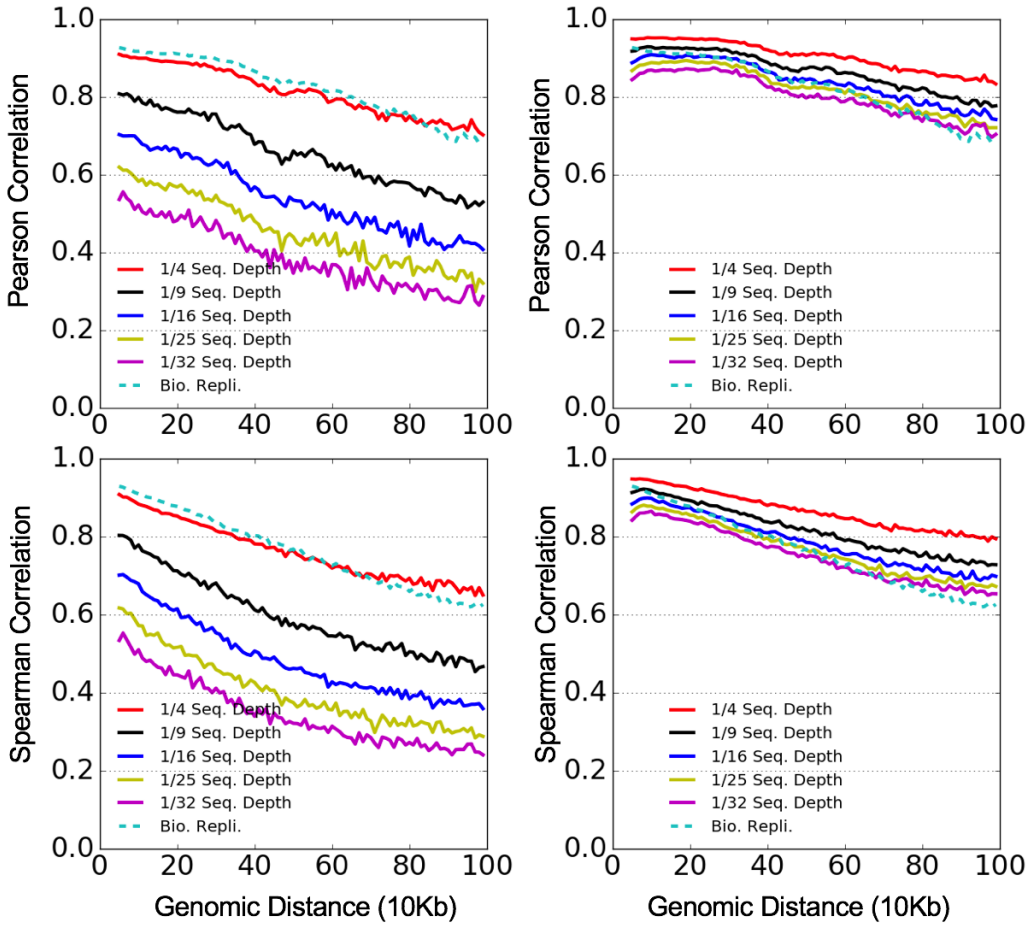


Figure 3.10: HiCPlus can generate high quality interaction matrix using a fraction of the original sequencing depth. Figures on the left column describe the correlations between down-sampled interaction matrices vs. the original high-resolution matrix. Figures on the right column describe the correlations between HiCPlus enhanced interaction matrices vs. the original high-resolution matrix. Compared with down-sampled matrix, HiCPlus significantly increased their correlation to the original deep sequenced data. We plot Pearson correlation coefficients in the top panels and Spearman correlation coefficients in the bottom panels.

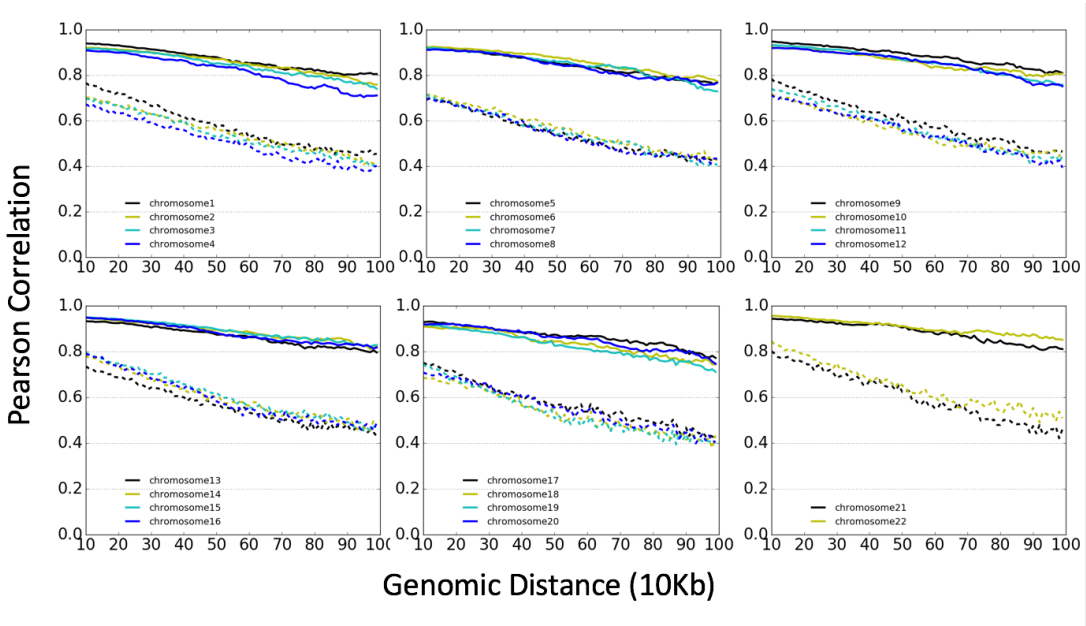


Figure 3.11: The performance of HiCPlus is stable on different chromosomes This figure shows the performance of the same model, which is trained on chromosomes 1-8, on all 22 chromosomes. The Pearson correlations on all chromosomes have nearly the same improvement comparing with the raw input sample.

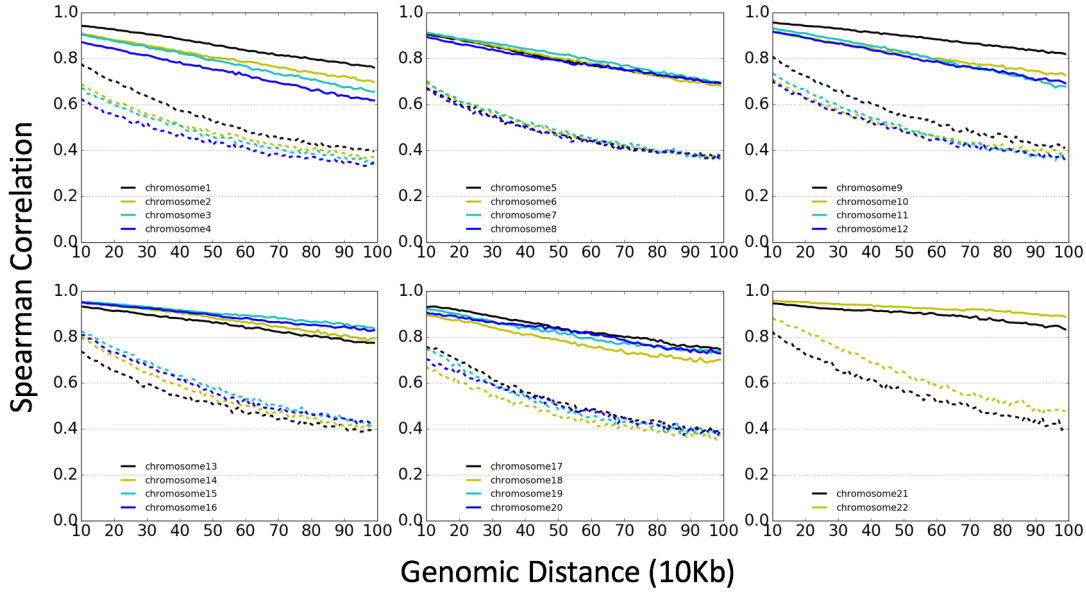


Figure 3.12: The performance of HiCPlus is stable on different chromosomes based on Spearman correlation This figure is the continue of Fig. 3.11, and Spearman ranking correlation is employed as the metrics in this figure.

resolution matrix, down-sampled matrix, and HiCPlus enhanced matrix at each genomic distance. As shown in Fig. 3.9b, HiCPlus enhanced matrix obtained much higher correlation with the real high-resolution Hi-C matrix than the down-sampled matrix at all genomic distances. Surprisingly, the correlations between the HiCPlus enhanced matrix and the real high-resolution Hi-C matrix are as nearly high as those between two real high-resolution matrices from two biological replicates in the same cell type (Fig. 3.9b, Fig. 3.10), suggesting that ConvNet framework can reconstruct a high-resolution interaction matrix using only a fraction of the total sequencing reads. To illustrate the performance of the model is consistent across different chromosomes, we test the performance of the same model, which is trained on chromosomes 1-8, on every chromosomes (as shown in Fig. 3.11 and Fig. 3.12). We didn't notice any difference in the improvements between the raw input and enhanced Hi-C.

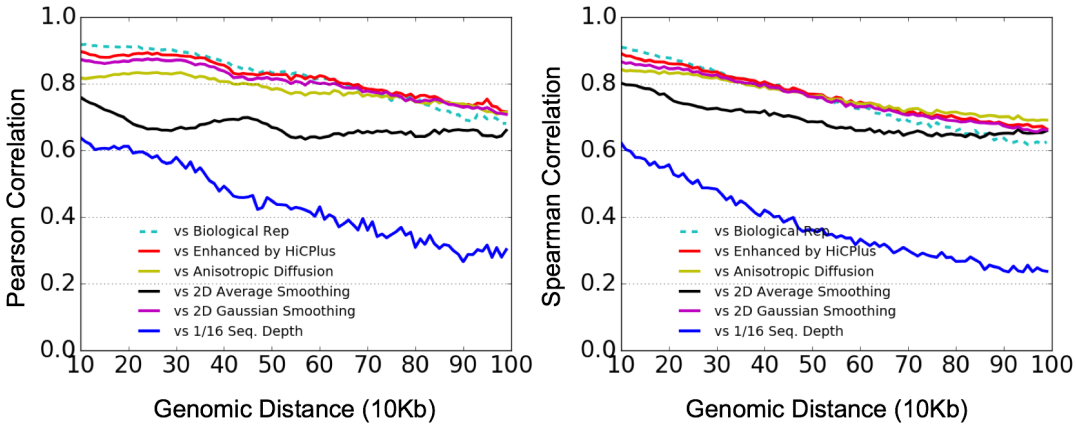


Figure 3.13: The performance of HiCPlus implemented by image denoising approach This figure is similar to Fig. 3 and we test several image-denoising approach. As shown the figure, all of the denoising approaches have some kind of the enhancement effect of Hi-C matrix but not as good as HiCPlus. Among the denoising approach, 2D Gaussian smoothing achieves much better result comparing with 2D averaging smoothing and Anisotropic diffusion. Therefore, in the following discussion, we are using the 2D Gaussian as representative of the image denoising approach for the baseline comparison.

To compare deep convolutional neural network with other potential approaches, we also implemented denoising-based method, including 2D Gaussian Smoothing, 2D

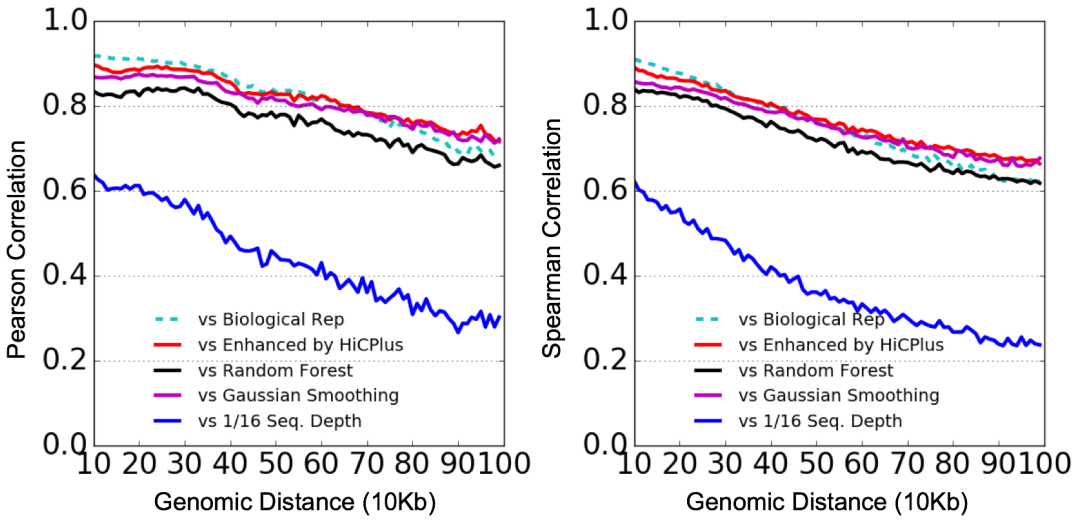
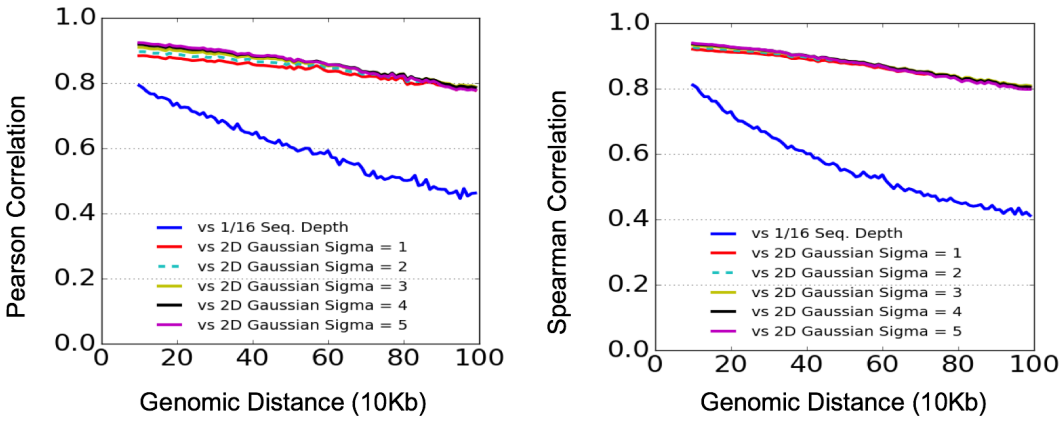


Figure 3.14: The performance of HiCPlus implemented by different models. This figure is similar to Fig. 3 and we present the performance of different approaches. As shown the figure, current version of HiCPlus implemented by convolutional neural network achieve the best performance. We also try the Supported Vector Regressor(SVR) but the performance is poor so we didn't plot in this figure.

average smoothing and anisotropic diffusion (Fig. 3.13) as well as other learning-based models (e.g. Random Forest Repressor(RFR) and Support Vector Repressor(SVR)) as shown in Fig. 3.14. The selection of parameters of 2D Gaussian Smoothing is shown in Fig. 3.15, and the parameters for 2D Average smoothing is obtained from Yang's work (T. Yang et al. 2017). We used default parameters for RFR and SVR in Sklearn (Jones, Oliphant, Peterson, et al. 2001–).

We also test the performance of HiCPlus on the normalized Hi-C matrix. It has been shown that there are systematic biases in Hi-C matrix (Yaffe and Tanay 2011; Hu et al. 2012), for example GC contents, number of cutter sizes, and mappability in each bin/region. Only after normalization, Hi-C data can be further analyzed to infer important features of 3D genome organization, such as determining the chromatin interactions between enhancers and promoters or identifying topologically associating domains (TADs). As shown in Fig. 3.16, the HiCPlus can be also employed to enhanced the normalized Hi-C matrix as long as the training sets are also normalized



Sigma	1	2	3	4	5
Average Pearson Correlation	0.8391	0.8486	0.8571	0.8608	0.8606
Average Spearman Correlation	0.8685	0.8730	0.8761	0.8760	0.8736

Figure 3.15: Testing the parameter for the 2D Gaussian smoothing This figure is similar to Fig. 3, and we choose the optimal parameter for the 2D Gaussian Smoothing, which is one of the baselines in this study. To determine the optimal parameter (the deviation, donating as Sigma) in the 2D Gaussian smoothing, we run the 2D Gaussian smoothing with different Sigma values. To quantitatively compare the correlation, we also list the average correlation in the distance 10-100 bins as shown the in the table. The performance of sigma = 3, 4, and 5 are very similar, and we pick sigma=4 as the optimal Gaussian kernel parameter for the following study. The study is performed at chromosome 9.

matrix tested whether HiCPlus works with normalized Hi-C data and the results show that HiCPlus can also enhance the resolution of normalized Hi-C data (Fig. 3.16).

3.3.3 Enhancing Hi-C interaction matrices across different cell-types

A key application for HiCPlus is to enhance the resolution of existing low-resolution Hi-C matrices from the previous studies (J. R. Dixon et al. 2012; J. R. Dixon et al. 2015; J. Fraser et al. 2015; Nagano et al. 2015; Jin et al. 2013; Leung et al. 2015; Seitan et al. 2013; Shen et al. 2012; Tang et al. 2015; Sofueva et al. 2013) with the model trained on the cell types where high-resolution Hi-C data are available (Rao et al. 2014; Jin et al. 2013). The results can also be used to address whether the

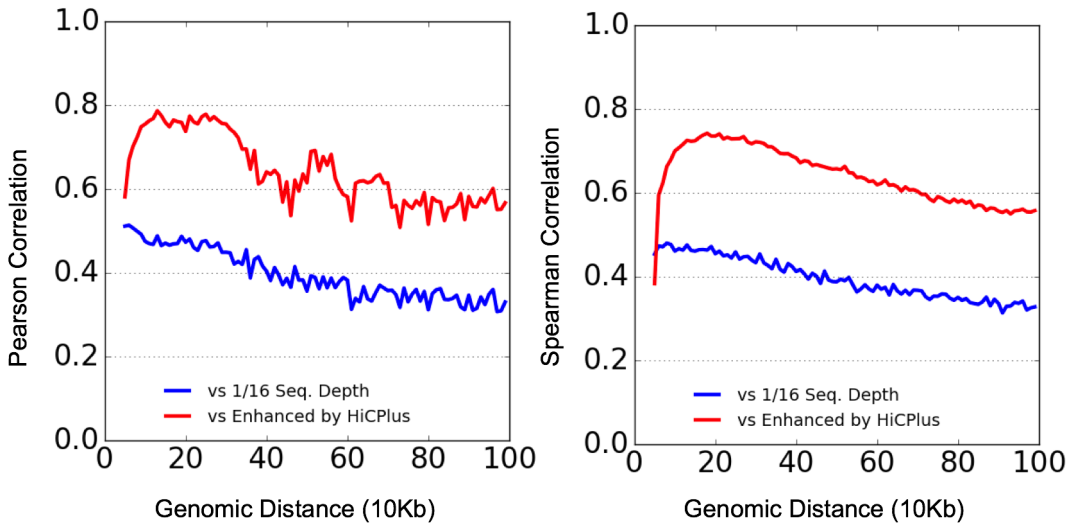


Figure 3.16: HiCPlus can also enhance normalized Hi-C interaction matrix HiCPlus model was trained and tested with ICE normalized Hi-C data in GM12878 cells at 10kb resolution.

low-level local patterns on Hi-C matrix are shared across different cell types as well. First, we trained the ConvNet model in three different cell types (GM12878, K562, IMR90) (Rao et al. 2014) with similar sequencing depths and tested their prediction performances in K562 cells. Similar to the procedure showed in the previous section, we down-sampled Hi-C reads in K562 to 1/16 and then applied ConvNet to enhance its interaction matrix. As shown in Fig. 4a, the enhanced Hi-C matrices using three different training data sets are highly similar to each other. More importantly, all of them are also similar to the original high-resolution interaction matrix (Fig. 3.17a, Fig. 3.18), suggesting that the local patterns/features captured by ConvNet framework from different Hi-C matrices are highly similar and can be used to enhancing Hi-C matrix in other cell types.

To further validate this observation, we trained the ConvNet model in GM12878 cells and applied it to enhance Hi-C matrices in three different cell types (GM12878, K562, IMR90). Again, we found that the ConvNet enhanced Hi-C matrices are highly similar to the real high-resolution Hi-C matrices. An example is shown in

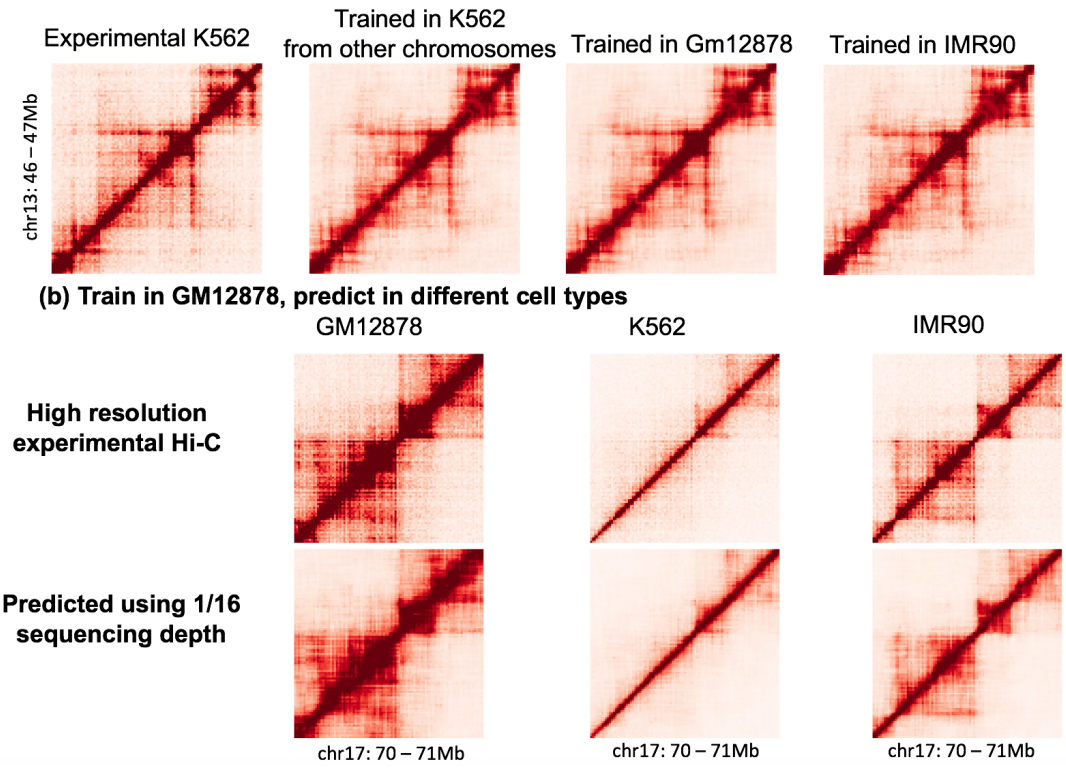


Figure 3.17: HiCPlus can learn model from one cell type and predict in other cell types Figures are real and HiCPlus enhanced matrices in GM12878, K562 and IMR90 at 10kb resolution. a, HiCPlus enhanced Hi-C matrices in K562 using models trained in three different cell types are highly similar to each other, and all of them are also similar to the original K562 interaction matrix. b, Model trained in GM12878 can be used to predict interaction matrices in different cell types (K562, GM12878 and IMR90)

Fig. 3.17b, where the chromatin interaction patterns in this region are highly dynamic across different cell types. However, the ConvNet enhanced matrices accurately depict these differences and help demonstrating these cell-type-specific TADs and chromatin interactions. Finally, we applied HiCPlus to systematically enhance 22 low-resolution Hi-C interaction matrices from 20 tissues/cell types generated in past several years (J. R. Dixon et al. 2012; J. R. Dixon et al. 2015; J. Fraser et al. 2015; Nagano et al. 2015; Jin et al. 2013; Leung et al. 2015; Seitan et al. 2013; Shen et al. 2012; Tang et al. 2015; Sofueva et al. 2013).

To predict such datasets, in the first step, we trained models for different sequencing depth from ($\times 4$ to $\times 16$). Then we generate the 10kb Hi-C interaction matrix from

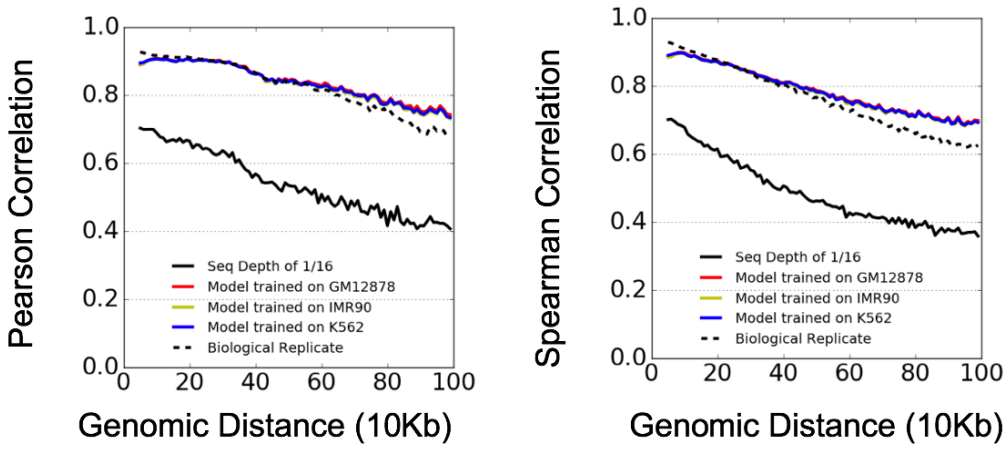


Figure 3.18: Quantitatively evaluation of the performance between the models trained on different cell types High correlations between predicted HiCPlus enhanced matrices using models trained in three different cell types and high resolution Hi-C at each genomic distance.

the BAM file in Hi-C library. In order to determine the enhancement scale, we calculate the ratio of the effective sequencing depth between the candidate Hi-C matrix and Hi-C training matrix between genomic distance of 25,000 to 1,000,000 base pairs. If the sequencing depth of candidates' Hi-C matrices is less than 1/16 of training Hi-C matrix, we use the $\times 16$ model.

3.3.4 Identification of meaningful interactions with HiCPlus enhanced matrices

It has been shown that strong chromatin interactions (loops) are enriched for important regulatory elements, such as enhancers and promoters (Rao et al. 2014). After demonstrating that HiCPlus can transform low-resolution Hi-C data to high-resolution Hi-C interaction matrix, we investigated whether these enhanced high-resolution matrices can facilitate the identification of meaningful chromatin interactions. For this purpose, we used Fit-Hi-C (Ay, Bailey, and Noble 2014) software, which can adjust random polymer looping effect and estimate statistical confidence of intra-chromosomal interactions. For the down-sampled Hi-C, there are two possible

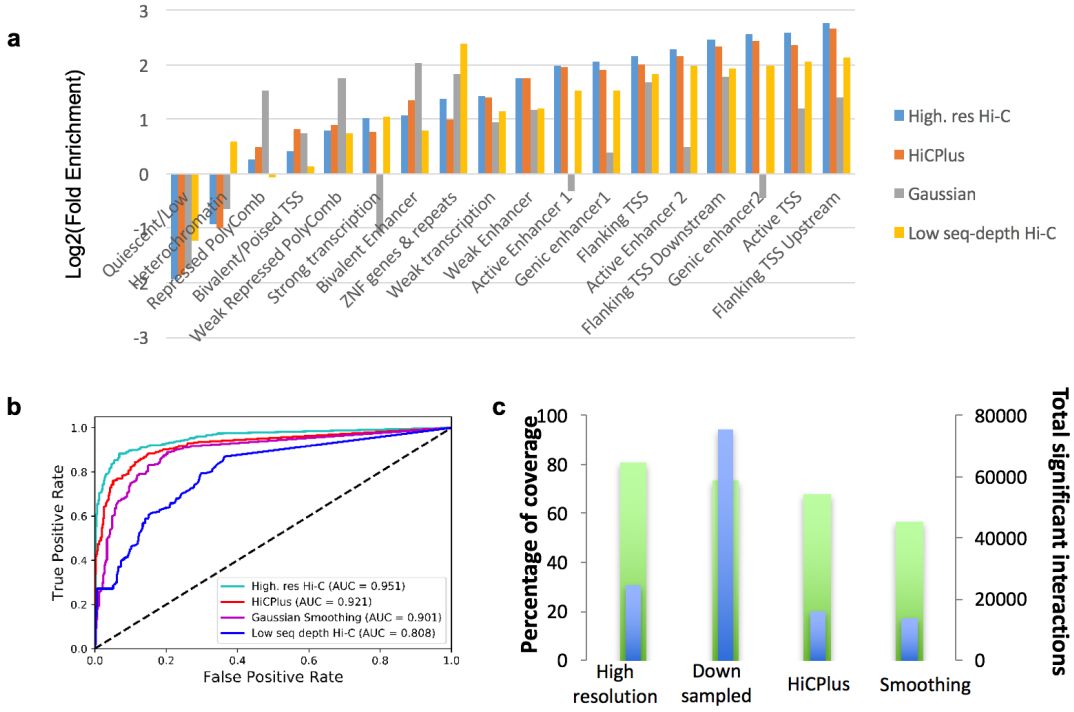


Figure 3.19: Identification of meaningful interactions with HiCPlus enhanced matrices a, Enrichment of potential functional element in predicted interacting regions, from down-sampled Hi-C, HiCPlus enhanced and real high-resolution Hi-C matrices in K562 cell line at 10kb resolution. The functional annotations are from chromHMM. The interaction regions are identified with Fit-Hi-C (cutoff of q-value $<$ 1e-06). ChromHMM states enrichment in those regions were shown as log2(fold-change) against interactions in whole-genome. b, ROC analysis of interactions from CTCF ChIA-PET with identified interacting peaks from down-sampled Hi-C, HiCPlus enhanced and real high-resolution Hi-C matrices in K562 cell line. c On the left axis, we plot the percentage of CTCF ChIP-PET identified chromatin interactions that are also detected by Fit-Hi-C from Hi-C matrices. On the right axis, it is the total number of interactions called by Fit-Hi-C for different Hi-C matrices

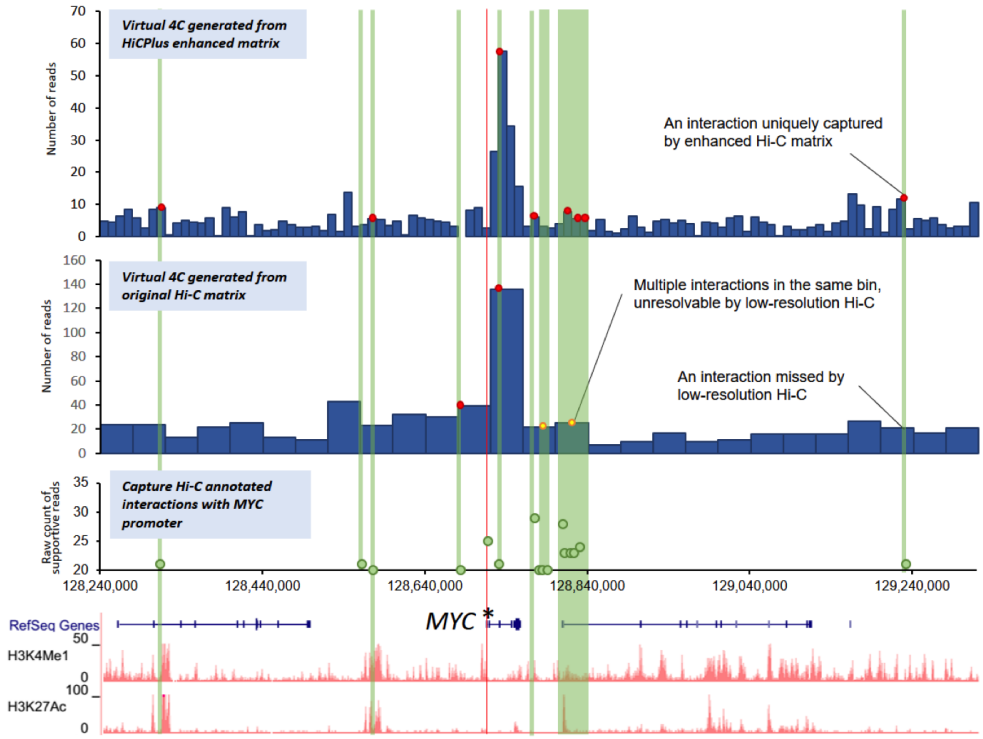


Figure 3.20: HiCPlus enhanced matrix captures significant interactions between MYC promoter and cis-regulatory elements that are missed or unresolved by low-resolution Hi-C matrix The top two virtual 4C tracks are generated using HiCPlus enhanced matrix (10kb resolution) and the original matrix (40Kb resolution) from Aorta tissue, anchored on MYC promoter (marked by *). We compared virtual 4C tracks with Capture Hi-C data surrounding MYC promoter, supported by at least 20 reads in GM12878 cells. Red dots indicate the Capture Hi-C peaks that are also detected by Hi-C. We notice that multiple Capture Hi-C interactions are mapped to the same 40kb bin and thus unresolvable by the low-resolution Hi-C matrix (yellow dots in the low-resolution virtual 4C). However, these interactions are captured by the HiCPlus enhanced matrix. We also notice that these interactions are between MYC promoter and potential distal enhancers, marked by H3K4me1 and H3K27ac.

way to run Fit-Hi-C: 1) run on low interaction numbers; 2) multiple the downsampled ratio to make the overall interaction intensities similar to the original high-resolution matrix. We run both of the methods and for method one, nearly no significant interactions are called because the value is too low. Therefore, unless otherwise noted, we run Fit-Hi-C based on the second method.

We applied Fit-Hi-C to the real high-resolution, 1/16 down-sampled, and HiCPlus enhanced interaction matrices at 10kb resolution in K562 cell line, respectively. We kept the predicted significant interactions (q-value $<$ 1e-06 as suggested in the manual) in genomic distance from 30Kb to 500Kb for further comparative analysis. Then we investigated whether the predicted chromatin interactions from three matrices are enriched for potential functional elements annotated by ChromHMM (Ernst and Kellis 2012).

As shown in Fig. 3.19a, significant interactions from the real high-resolution Hi-C matrix and HiCPlus enhanced matrix show similar patterns: enriched for active states, such as enhancer-associated states (Weak Enhancer, Active Enhancer 1&2, Bivalent Enhancer and Genic enhancer1&2) and promoter-associated states (Flanking TSS Upstream, Flanking TSS Downstream and Active TSS), while depleted of inactive states, such as quiescent and heterochromatin-associated states (Quiescent/Low and Heterochromatin). On the contrary, the interactions identified in the down-sampled Hi-C matrix show discrepant patterns with those identified in real high-resolution Hi-C matrix. For example, they are enriched for heterochromatin and minimal if any enrichment of active TSS, suggesting that interactions identified from the down-sampled matrix are of poor quality and might give false information if analysed at this resolution.

Next, we compared the predicted chromatin interactions from the real high-resolution Hi-C, down-sampled Hi-C, and HiCPlus enhanced matrices, with the identified chromatin loops by CTCF ChIA-PET in the same cell type. We used the

identified CTCF mediated chromatin loops from ChIA-PET as true positives. As for negatives, we randomly selected the same number of pairs of CTCF binding sites that are not predicted as interacting pairs by ChIA-PET (methods). Then we plotted the AUC (Area Under the Curve) to evaluate the performance of our model.

As shown in Fig. 3.19b, CTCF interacting pairs and non-interacting pairs are separated in the predicted results from HiCPlus enhanced matrix (average AUC = 0.85). We also observed that the AUC score for HiCPlus enhanced matrix is significantly higher than the AUC from down-sampled matrix (p-value < 0.05). Finally, we compared the overlap between significant interactions identified in all three interaction matrices with the ChIA-PET identified interactions (Fig. 3.19c). We use 5% false discovery rate (FDR) cutoff of Fit-Hi-C to call the significant interactions from Hi-C matrices, and we define the coverage as the percentage of interactions dtect by ChIP-PET which are also significant interactions in Hi-C. Although the low-sequenced Hi-C has slightly higher coverage but the total number of the significant interactions is too large, indicating high rage of false positive. HiCPlus and smoothing results have similar number of the significant interactions, and HiCPlus has higher coverage .

To further show the power of HiCPlus framework, we applied it to enhance the Hi-C data set from Aorta tissue where only low-resolution (40kb) matrices are available (Fig. 3.20). By comparing chromatin interactions from Capture Hi-C, we observe that HiCPlus enhanced matrix captures significant interactions between MYC promoter and cis-regulatory elements that are missed or unresolved by low-resolution Hi-C matrix. For example, multiple Capture Hi-C interactions are mapped to the same 40kb bin and thus unresolvable by the low-resolution Hi-C matrix (yellow dots on the second 4C track). However, these interactions are captured by the enhanced matrix, suggesting that HiCPlus can improve the resolution of Hi-C interaction matrix and reveal meaningful interactions that are missed by original low-resolution Hi-C data.

In summary, the ConvNet framework can significantly improve the quality of in-

teraction matrix for insufficiently sequenced Hi-C samples and further facilitate identifying biologically meaningful interactions that are enriched for potential functional elements and validated by other techniques.

3.4 Discussion

Here we present HiCPlus, the first computational approach to infer high-resolution Hi-C interaction matrices from low-resolution Hi-C data. Our framework can construct the interaction matrix with similar quality, but using only 1/16 or even fewer sequencing reads. We systematically applied HiCPlus to generate high-resolution matrix for 22 tissue/cell types where only low resolution Hi-C data are available, covering a large variety of human tissues.

We observe that Hi-C interaction matrices are composed of a series of low-level repeating local patterns, which are shared across all cell types and tissues. These features can be effectively captured by our ConvNet framework and used to enhance Hi-C matrix in different cell types.

It is interesting to further study the patterns utilized by the model to enhance the Hi-C matrix. Considering that the simple Gaussian smoothing could also achieve the pretty large increase in the Hi-C quality (Fig. 3.13) and 2D Gaussian smooth can also be regarded as the convolutional operation, we believe filtering out the noise by the smoothing should a significant way to remove the false positive interactions. The pattern utilized by the smoothing can be described as *the point should be similar to its nearby points, and if a strong interaction point is isolated in low interaction regions, it is more likely to be a noise signal*.

In Fig. 3.21, we compare the overlap significant interactions called by the Fit-Hi-C between different version Hi-C with experimental high-resolution Hi-C. As expected, when the sequencing depth is low, the noise level is very high, and a large number of false positives are observed. Both Gaussian smoothing and HiCPlus can filter out

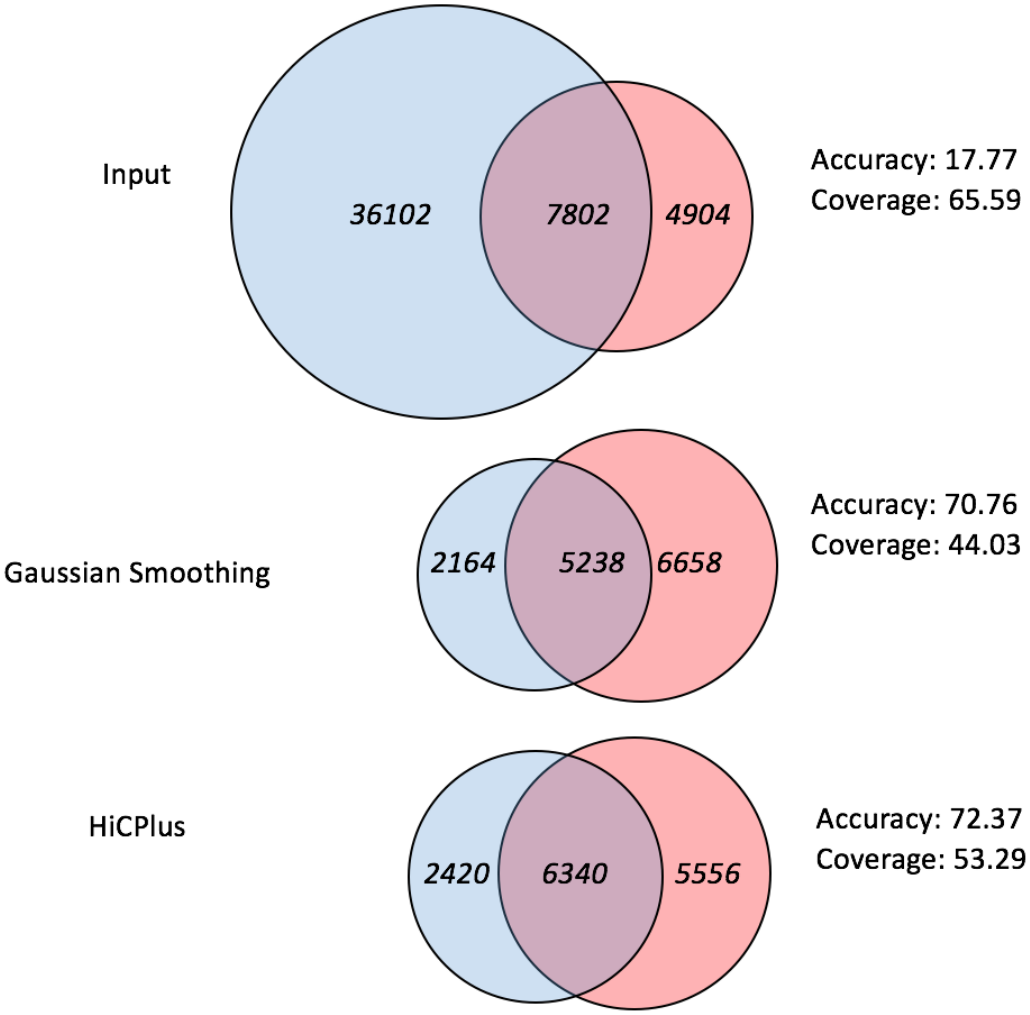


Figure 3.21: Overlap on the Fit-Hi-C significant interactions with experimental high resolution Hi-C We use the Fit-Hi-C to call the significant interactions on raw input, smoothed and HiCPlus enhanced Hi-C matrix and count the overlap with the experimental high-resolution Hi-C. We use p-value of 10e-6 for the threshold as suggested by Fit-Hi-C. The raw version of downsamples input Hi-C cannot detect any significant interactions so we multiple the down sample rate (16) to make all Hi-C matrix have the similar overall intensity. The number of significant interaction in the input Hi-C is more than 3 times as the high-resolution Hi-C, which is regarded as the ground truth, indicating the high noise level in the insufficient-sequenced Hi-C. The Hi-C matrix with Gaussian smoothing has much less significant interaction. Comparing with Gaussian smoothing, the HiCPlus, have similar number of significant interactions and better performance in both accuracy and coverage. We believe the multi-layer non-linear filtering in convolutional neural network to distinguish noise and real signals.

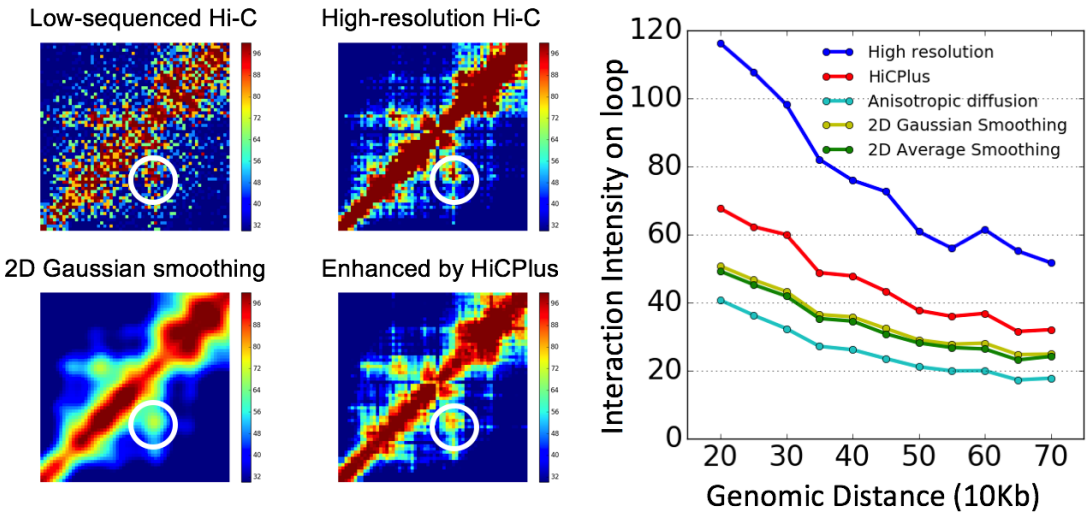


Figure 3.22: The HiCPlus outperform simple 2D smoothing at important regions Besides the the analysis on the entire genome, we also investigate the performance on the loop peak, which is one of key patterns of Hi-C interaction heatmap. We plot the Hi-C heatmap with high contrast on the left. The Gaussian smoothing works good for noise reduction, however, it also reduce too much signal on the loop peak(as shown in the white circle). On the HiCPlus, the noise is also removed and the strong interaction peak is also conserved as well. We believe that smoothing is an important operation of convolutional neural network in HiCPlus to remove excess noise. Comparing with simple smoothing, the multiple steps of non-linearity filters in the HiCPlus enable HiCPlus to learn more complicated features from the train data sets. For example, in the loop peaks here, from the biological knowledge, we can describe as at the top of the Topological Association Domains(TADs), the strong interacted bins are more likely to be a true signal of the loop peak rather than random noise. The simple smoothing is unable to distinguish loop peaks and noise.

noise, and HiCPlus outperform the 2D Gaussian smoothing in both accuracy and coverage, indicating the HiCPlus can capture more complicated features comparing with simple smoothing.

We further compare the performance between Gaussian smoothing and HiCPlus on loop region, which is one of the most important discoveries in Hi-C (Rao et al. 2014) , and the result is shown in Fig. 3.22. Loop peaks are strong interaction peak comparing with the background, and the results show the Gaussian smoothing reduce the signal too much while HiCPlus more efficiently distinguish such loop peaks with noise. However, most of these local patterns are still represented as black boxes

in the intermediate convolutional layers and therefore are not human interpretable. We hypothesize that these features are related to important functions in 3D genome organization, such as chromatin loops and TADs. More work on visualizing and interpreting these features are imperative and will be of great values to deepen our understanding of the high-order genome organization and gene regulation.

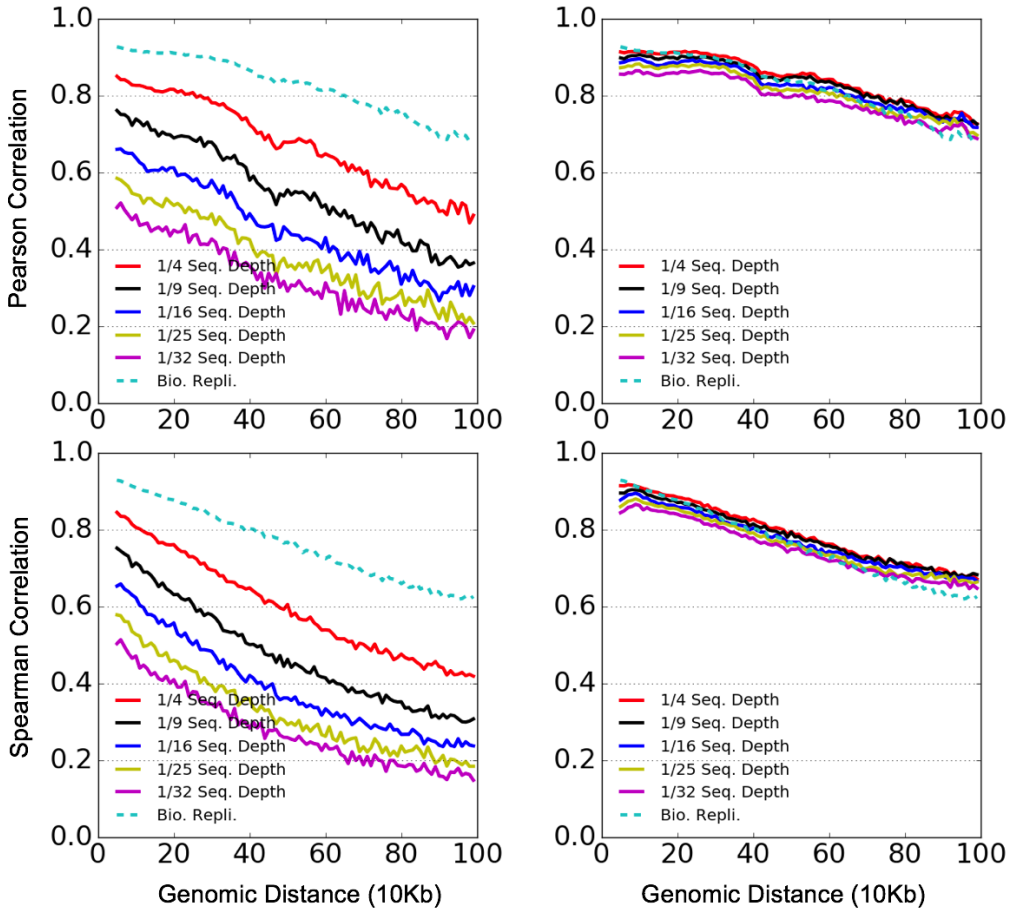


Figure 3.23: We observe that HiCPlus enhanced matrix is also highly similar to the interaction matrix from other biological replicate in the same cell type (GM12878).

Another caveat is the ground truth used for training and evaluating in the ConvNet framework. Throughout the analyses in this work, we used the real high-resolution Hi-C matrix as the ground truth/gold standard. However, there are natural variations even between high-resolution interaction matrices from different biological replicates in the same cell type. Intriguingly, we found that in some cases, the Con-

vNet enhanced matrix is more correlated with the other biological replicate than with the replicate where it is trained (Fig. 3.23). Further, in the functional enrichment analysis (Fig. 3.19a), the significant interactions in the ConvNet enhanced matrix are highly enriched for most of the epigenetic markers than those from the real high-resolution Hi-C matrix. In addition, previous work from other disciplines (Goodfellow, Bengio, and Courville 2016; Srivastava et al. 2014; Sukhbaatar and Fergus 2014) have reported that introducing noises in the training process can increase the prediction accuracy of the deep learning model. It is possible that the deep ConvNet model can distinguish noises and real signals in the Hi-C matrices, which contributes to the interaction matrix enhancement. Further investigations are needed to validate and interpret these interesting observations and the results might shed light on how to improve the computational model and deepen our understanding of chromatin interactions. We want to point out that sequencing depth has great impact on the performance of HiCPlus. In this work, to make enhanced matrices for the 20 human tissue Hi-C data(22 replicates), we trained three different models according to their available sequencing depth: $>$ 80 million, 50-80 million, $<$ 50 millions (more detailed breakdown in Table. 3.1. To achieve the best result, an individual user is recommended to retrain the model according to the sequencing depth. The user can simply down-sample the Hi-C reads in GM12878 or IMR90 to match their read numbers and run our pipeline to train their model.

In summary, HiCPlus presents the first deep learning framework for enhancing the resolution of Hi-C interaction matrices. By leveraging interaction frequencies from neighbouring regions and learning regional patterns from available high-resolution Hi-C data, HiCPlus can generate high-resolution Hi-C interaction matrices at a fraction of the original sequencing reads. With the fast accumulation of Hi-C data in different cell lines and tissue types, we provide a rich resource and a powerful tool for the study of 3D genome organization and gene regulation.

Table 3.1: Difference in the change of the distances D caused by different occlusions as shown

Dataset	Tissue/Cell type	Reads in region	Enhance Ratio	Model	total reads
Muscle_Psoas(PO3)	Psoas	2.0	71.0	16	9.3
Spleen(SX3)	Spleen	2.7	51.7	16	19.3
Pancreas(PA3)	Pancreas	2.9	49.3	16	11.6
Lung(LG2)	Lung	3.6	39.9	16	20.3
Lung(LG1)	Lung	3.7	38.6	16	13.3
Muscle_Psoas(PO1)	Psoas	5.1	28.0	16	20.0
Ovary	Ovary	5.4	26.1	16	27.7
Bowel_small	Small Bowel	5.7	24.8	16	25.9
Pancreas(PA2)	Pancreas	7.4	19.2	16	27.4
AdrenalGland	Adrenal	8.4	16.9	16	28.8
Spleen(SX1)	Spleen	8.5	16.6	16	48.1
Bladder	Bladder	9.3	15.3	16	39.6
cortex_DLPFC	Dorsolateral Prefrontal Cortex	9.9	14.4	16	33.4
Hippocampus	Hippocampus	10.6	13.4	16	38.0
H1-NPC(rep1)	Neural Progenitor Cell	14.8	9.6	9	50.3
VentricleRight	Right Ventricle	18.7	7.6	9	60.4
H1-NPC(rep2)	Neural Progenitor Cell	19.2	7.4	9	70.6
Aorta_STL002	Aorta	30.2	4.7	4	101.6
H1-TRO(rep1)	Trophoblast-like Cell	35.2	4.0	4	88.7
H1-TRO(rep2)	Trophoblast-like Cell	42.9	3.3	4	105.7
Liver_STL011	Liver	49.9	2.8	4	161.1

Chapter 4

Training the Denoising Network without Clean Datasets and Its Application to Hi-C

4.1 Introduction

Denoising (a.k.a noise reduction) is a process to reduce the noise from raw data. Experimental data is usually the mixture of signal and noise, thus reducing noise from experimental data is required (Kantz et al. 1993) to enhance data quality. In a broader view, a dataset is a corrupted version of the clean signal, where 'corruption' is the addition of noise. As a result, the denoising can also be regarded as a process to restore the clean version of signals from the corresponding corrupted noisy version.

4.1.1 Learning-based methods to reconstruct corrupted data

The learning-based approach does not require prior knowledge about the mechanism of the corruption process (e.g. the distribution of the noise). Instead, it learns strategies to reverse the corruption automatically from corrupted/uncorrupted data pairs.

The general procedure of learning-based data restoration includes:

1. Obtain the corrupted and uncorrupted version of data $\tilde{X}$ and X, respectively.
2. If the corrupted version $\tilde{X}$ doesn't exist naturally, simulate the corruption process $C(X|\tilde{X})$ to generate the corrupted data sets $\tilde{X}$.

3. Learn the model F, which reflects the generalized mapping relationship between the uncorrupted X and corrupted $\tilde{X}$. The aim of the training process is to minimize the dissimilarity $L(X, F(\tilde{X}))$.

4. Apply the model to data sets where uncorrupted version is unavailable, to reconstruct the corresponding uncorrupted version $F(X)$.

In recent years, with the development of the deep neural network (LeCun, Bengio, and G. Hinton 2015; Goodfellow, Bengio, and Courville 2016; Schmidhuber 2015), most learning-based data reconstruction approaches employ multi-layer neural networks as the model to reconstruct the uncorrupted data from the corrupted version in multiple fields, including image (Dong et al. 2016; Burger, Schuler, and Harmeling 2012; Eigen, Krishnan, and Fergus 2013), audio (Maas et al. 2012), and experimental data (Koh, Pierson, and Kundaje 2017; Y. Zhang et al. 2017).

4.1.2 Experimental data enhancement

Several advances have made in using the deep neural network to enhance the experimental data, mostly in the field of computational biology (Koh, Pierson, and Kundaje 2017; Y. Zhang et al. 2017). In this area, high-quality data is only available in very limited types of samples due to the cost, while low-quality data is much easier to obtain. These approaches employ a similar basic procedure as discussed above, where low-quality data sets are regarded as the corrupted version, and high-quality data sets are approximately considered as the golden standard. The model is trained with high-quality and low-quality data pairs and applied to the samples where only low-quality data sets are available. The limitation of these methods is that the high-quality data is still not perfectly clean, limiting the quality of the enhanced data.

4.1.3 Our Contribution

All of the approaches above need the clean data in some samples to train the denoising model in a supervised manner; then the model can be applied to those samples where the ground truth is unavailable. However, because of the limitation of experiments, it is nearly impossible to obtain noise-free data in most domains to train the denoising network. Conventionally, to obtain a higher quality of experimental data, a standard approach is to run parallel experiments under identical condition multiple times to generate several copies (replicates) of the data. The final result is thus the average of all the parallel experimental results, which may be better than any of the individual data, although this approach requires more resources.

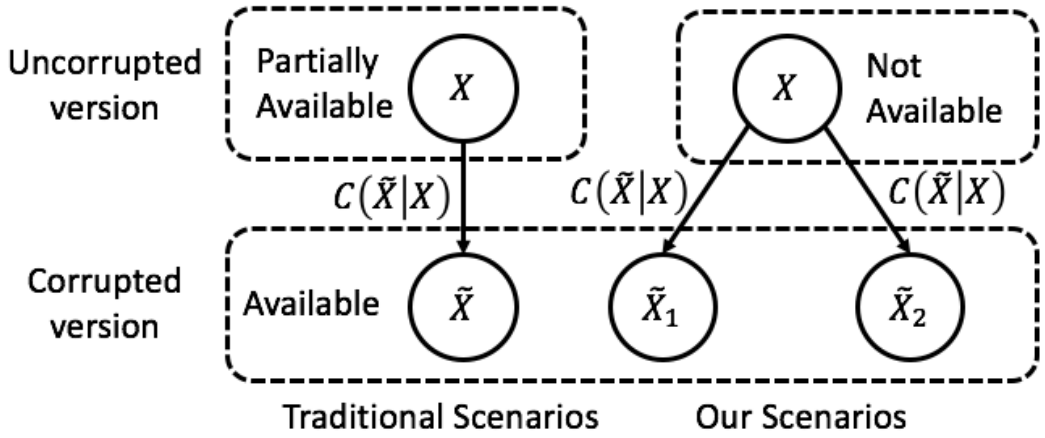


Figure 4.1: The application scenarios in our denoising network. X is the clean data, and $\tilde{X}$ is an observation (experimental data) from the ground truth X. The process to obtain the $\tilde{X}$ from X can be regarded as a corruption process which is presented as $C(X|\tilde{X})$. On the left, we plot the traditional scenarios of the reconstruction, where the uncorrupted data are available on the part of the data sets. On the right, it is the scenarios discussed in this work, where none of the uncorrupted data sets is available.

In this paper, we propose a framework to train the model for reducing noise in experimental data without uncorrupted data. The training of the model utilizes the data sets from two (or more) parallel experiments under identical conditions. In Fig. 4.1, the left part represents application scenario of previous denoising networks

as discussed above. In those work, the uncorrupted data(e.g. clean image) is at least partially available and can be used as training data. In the right part of Fig. 4.1, we illustrate the scenario of our work when the uncorrupted ground truth is not available, which reflects most cases in experimental science.

Basic idea of our approach In experimental science, the observed experimental data $\tilde{X} = X + \epsilon$, where X is the true signal and ϵ is the noise. The signal X is stable and consistent across all parallel experiments, while the noise ϵ is volatile and will change across parallel experiments. Our approach takes advantage of these properties to train denoising network to distinguish between the true signal X and the noise ϵ.

4.1.4 Hi-C experiments

Hi-C (Lieberman-Aiden et al. 2009) is a technique to detect the 3D structure of genomes; the output of a Hi-C experiment is a 2D heatmap(Fig. 4.6a) which provides the interaction intensity between all loci on the genome. This heatmap provides important information about the 3D proximation between the loci, in particular for those who are far away along the DNA string, and such loci usually contain important functional units of the genome (Rao et al. 2014).

In the Hi-C experiment, a chemical reagent is added to crosslink the DNA string, and the loci closing to each other have the higher probability to be crosslinked. Then, the long DNA string is digested into small pieces to sequence. The value of each pixel of a Hi-C heatmap is the number of small DNA sequencing (a.k.a DNA reads) whose two ends binding two different loci by the cross-linking operation, and reflects the probability of crosslinking between two loci. Therefore, to enhance the signal-to-noise ratio of Hi-C, more short DNA sequencing reads are needed. However, since the cost of sequencing is high, the total number of DNA sequencing reads cannot increase infinitely. Even in the best Hi-C datasets obtained so far (Rao et al. 2014), two identical biological replicates cannot match each other exactly, indicating the

existence of noise in Hi-C data, making it precious to reduce the noise level in Hi-C data.

4.2 Model Description

4.2.1 Siamese Network

Siamese network is a class of network structure, in which two or more identical sub-structures share the same set of parameters (Bromley et al. 1994). Siamese network has achieved great success in a series of verification and identification problems including signature (Bromley et al. 1994), face (Chopra, Hadsell, and LeCun 2005; Khalil-Hani and Sung 2014; L. Zheng et al. 2016; Bianco 2017), gait (C. Zhang et al. 2016) and voice (Sandouk and K. Chen 2016; Kamper, W. Wang, and Livescu 2016).

The key of Siamese network is to process the raw form of comparable sample pairs (X_1, X_2) by multi-layer neural networks $F(X, \Theta)$ with shared topology and parameters Θ, and generate the representation of the sample pairs (Z_1, Z_2), where $Z_1 = F(X_1, \Theta)$, $Z_2 = F(X_2, \Theta)$ and Θ is the shared parameter in the multi-layer neural network. Then, the similarities $\mathcal{D}$ of the sample pairs are evaluated in the representation form as $\mathcal{D}(Z_1, Z_2)$.

In these verification problems, the label Y can be defined based on the relationship of sample pairs (X_1, X_2). For instance, in face verification problem, the binary labels for the image pairs have been defined as 0 if the image pair is from the same person or 1 if the image pair is from different persons (Chopra, Hadsell, and LeCun 2005). In the following step, the loss can be calculated based on Eq. 4.1.

$$\mathcal{L}(X_1, X_2, Y) = L(\mathcal{D}(Z_1, Z_2), Y) \tag{4.1}$$

where L is the loss function,and $\mathcal{D}$ is the dissimilarity.

4.2.2 Our model

Here, we present a denoising Siamese network(DSN), which utilizes the Siamese topology to train a denoising network $F(\Theta)$. The aim of denoising network $F(\Theta)$ is to obtain clean data Z from noisy experimental data $\tilde{X}$ (Eq. 4.2).

$$Z = F(\tilde{X}, \Theta) \tag{4.2}$$

where X is the noisy raw data, and Z is the denoised data.

Siamese Network Topology

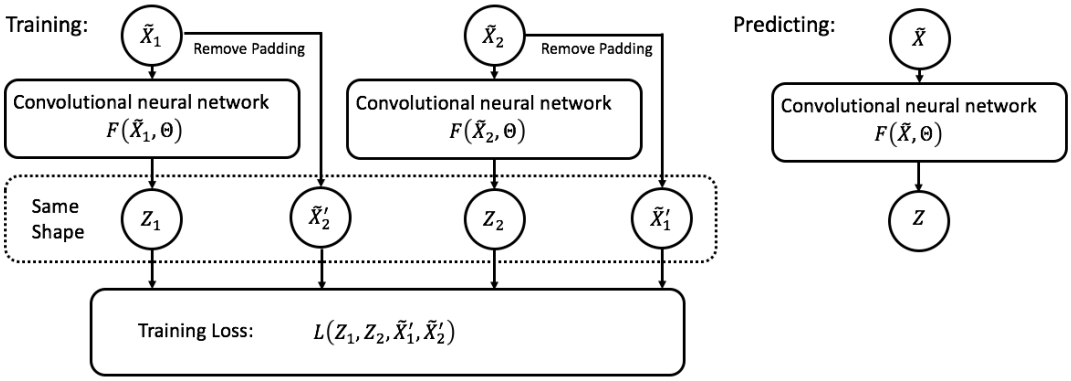


Figure 4.2: The network topology of our denoising siamese network. $\tilde{X}_1$ and $\tilde{X}_2$ are two independent experimental observations for the same sample from the identical experimental procedure. Z_1 and Z_2 are obtained from $F(\tilde{X}, \theta)$, which is the convolutional neural network with shared parameters θ. Since Z_1 and Z_2 have shrunk size after the convolutional operation, $\tilde{X}_1$ and $\tilde{X}_2$ will remove the padding region to obtain the $\tilde{X}_1'$ and $\tilde{X}_2'$ which are in the same shape with Z_1 and Z_2. Then Z_1 , Z_2 , $\tilde{X}_1'$, $\tilde{X}_2'$ are employed to calculate the loss. The purpose the training stage is to obtain the optimal parameters Θ of denoising network F. In the testing stage, only $F(\Theta)$ is needed to perform the denoising operation.

Unlike the denoising network discussed above, no uncorrupted data Z set is available to train the network $F(\Theta)$ in a supervised way, and we can only exploit the data sets from two identical experimental procedure (two replicates). Assuming an ideal denoising network $F(\Theta)$ can remove all of the noise generated during the experimental process, and X_1 and X_2 are two data sets from two experimental replicates of the

same sample from the same experimental procedure, $F(X_1, \Theta)$ and $F(X_1, \Theta)$ should be identical since data sets $\tilde{X}_1$ and $\tilde{X}_2$ reflect the same clean data.

The topology of the denoising Siamese network is shown in Fig. 4.2. In the training process, the input is two experimental replicates $\tilde{X}_1$ and $\tilde{X}_2$ for the same type of samples. The convolutional neural network is employed as the denoising network $F(\Theta)$ and Θ is the parameters (weight and bias) in the network. After training, only $F(\Theta)$ is needed to apply to the noisy data.

In training, to compare the proposed clean data Z_1 and Z_2, the raw data $\tilde{X}_1'$ and $\tilde{X}_2'$ is processed to the same dimension as Z_1 and Z_2 by removing the padding margin. In the following step, Z_1, Z_2,$\tilde{X}_1'$ and $\tilde{X}_2'$ is combined as the input for the loss function. In prediction, $F(\tilde{X}, \theta)$ is used to reconstruct clean data Z from corresponding corrupted version $\tilde{X}$

In the verification problem, to extract the key features, the dimension of the representation Z is usually much smaller than the raw form X. For example, in face verification (Chopra, Hadsell, and LeCun 2005), the raw form for each sample is 56 $\times$ 46 pixels, and representation for comparison is only a vector with a length of 50. In our project, we will keep Z's dimension the same with X except for the proper padding removal during the convolutional operation to calculate the loss.

Loss Function

A desired denoising network $F(\Theta)$ should have two properties:

- The denoised data sets $F(\tilde{X}_1, \Theta)$ and $F(\tilde{X}_2, \Theta)$ should be similar.
- The loss of the signal should be minimal

We use L_S to represent similarities between denoised data sets, and Euclidean distance is utilized as the metrics describe L_S (Eq. 4.3)

$$L_S = ||Z_1 - Z_2|| \tag{4.3}$$

where $Z_1 = F(\tilde{X}_1)$ and $Z_2 = F(\tilde{X}_2)$

To evaluate the loss of the signal during the denoising process, we propose the average of denoised data Z_1 and Z_2 should be the same with the average of raw data $(\tilde{X}_1, \tilde{X}_2)$. Therefore, we design the metrics L_D as shown in Eq. 4.4.

$$L_D = ||(Z_1 + Z_2) - (\tilde{X}_1' + \tilde{X}_2')|| \tag{4.4}$$

where $\tilde{X}_1'$ and $\tilde{X}_2'$ are raw data with padding region removed to match the dimension of Z_1 and Z_2.

Based on L_S and L_D, we define the loss function in Eq. 4.5.

$$L = \lambda L_S + L_D \tag{4.5}$$

where λ is the hyper-parameter to adjust the noise level to be removed.

In the loss function, the ratio between L_S and L_W is the hyper-parameter(λ). When the λ value is large, the denoising network $F(\Theta)$ will focus on removing noise to minimize L_S. On the contrary, when λ is small, the $F(\Theta)$ will keep the output Z_1 and Z_2 similar with their raw form $\tilde{X}_1'$ and $\tilde{X}_2'$, respectively.

Convolutional Neural Network

Convolutional Neural Network (CNN) has been proven to be effect to reconstruct uncorrupted data from corrupted data sets (Dong et al. 2014; Dong et al. 2016; Burger, Schuler, and Harmeling 2012; Koh, Pierson, and Kundaje 2017; Y. Zhang et al. 2017). Therefore, we employ the CNNs as the denoising network in this work. Our basic CNN structure is acquired from previous work (Dong et al. 2014; Dong et al. 2016; Y. Zhang et al. 2017), where three convolutional layers are utilized, and the output from the last convolutional layer is the final output without any fully

connected layers. We illustrate our CNN in Fig. 4.3. Comparing with the previous work (Dong et al. 2014; Dong et al. 2016; Y. Zhang et al. 2017), we add two noise layer to prevent the model just "copy" the noisy input data to the output. The selection of parameters in CNNs, including the filter number and filter size, will be discussed in the following section.

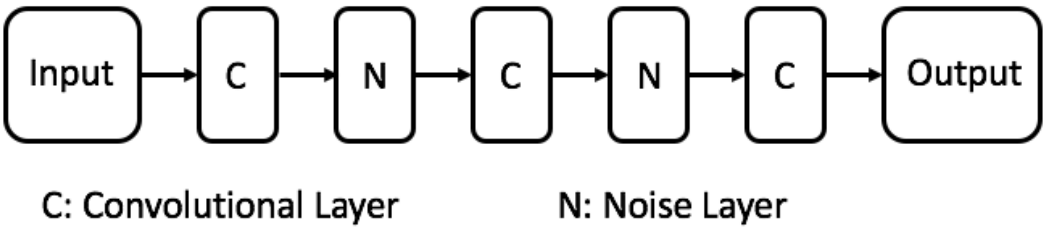


Figure 4.3: Structure of CNNs in siamese network. The output of last convolutional layer is also the output layers. The noisy layer using Gaussian dropout method as describe in (Srivastava et al. 2014)

Training of the network

As discussed in the section of the loss function, the training process is to minimize the L in Eq. 4.5, which is always a positive number. In our implementation, we simply set the target value for $Y = 0$, and use the standard backpropagation to obtain the minimal L according to Eq. 4.6.

$$argmin(L - Y)^2 \tag{4.6}$$

4.3 Evaluation on Simulated Noisy Data

To validate the performance of our approach, we test our method on simulated data. Since any data from experiments may contain noise, we choose the MNIST data sets of handwritten digits (LeCun et al. 1998) for our test. The 28×28 matrices of the MNIST input are employed as the clean data. To generate noisy data, noise is added to the clean data to simulate the data corruption process in an experiment. Finally,

our approach is applied to the noisy data, and the performance will be evaluated by comparing with the clean data.

4.3.1 Generating Noisy Data

The clean MNIST data is rescaled to range between 0 and 1. The simulated noisy data is generated in the following step:

1. Considering the matrix wound shrink during the convolutional operation, the width 6 white padding region is added to surrounding original 28×28 matrix , and 40×40 matrix is obtained.

2. The white Gaussian noise $\mathcal{N}(0, 0.5)$ is added each pixel in the matrix, including the newly added padding region.

3. To add the complexity of the noise, we set all values x on the noisy data matrix according to Eq. 4.7.

$$x = \begin{cases} 0 & \text{if } x < 0 \\ 1 & \text{if } x > 1 \\ x & \text{otherwise} \end{cases} \tag{4.7}$$

In the simulation process, the MNIST clean data sets X is independently processed twice by the procedure described above to generate two corrupted noisy data $\tilde{X}_1$ and $\tilde{X}_2$ in the training datasets. The clean data set X and the procedure would not be accessed in the model training process in any form.

The testing data is generated by the same procedure by adding noise to the clean data X to get $\tilde{X}$. The training sample size is set at 50000 and the testing sample size is set at 10000. There is no overlapping between training samples and testing samples.

4.3.2 Evaluation

We study the performance on different hyper-parameters in CNNs as discussed in the supplementary material. The result is not quite sensitive to hyper-parameters, and we determine the first size on each convolutional layer are 3×3, 7×7 and 5×5, respectively.

We utilize three metrics to evaluate the similarity between denoised data and original clean data: 1) Mean Squared Error(MSE), 2)average Pearson correlation for each sample, 3)Peak signal-to-noise ratio(PSNR). Since the original clean data in MNIST is range from 0 to 1, the denoised data is re-scaled to between 0 and 1 to calculate all of these metrics.

4.3.3 Impact of λ

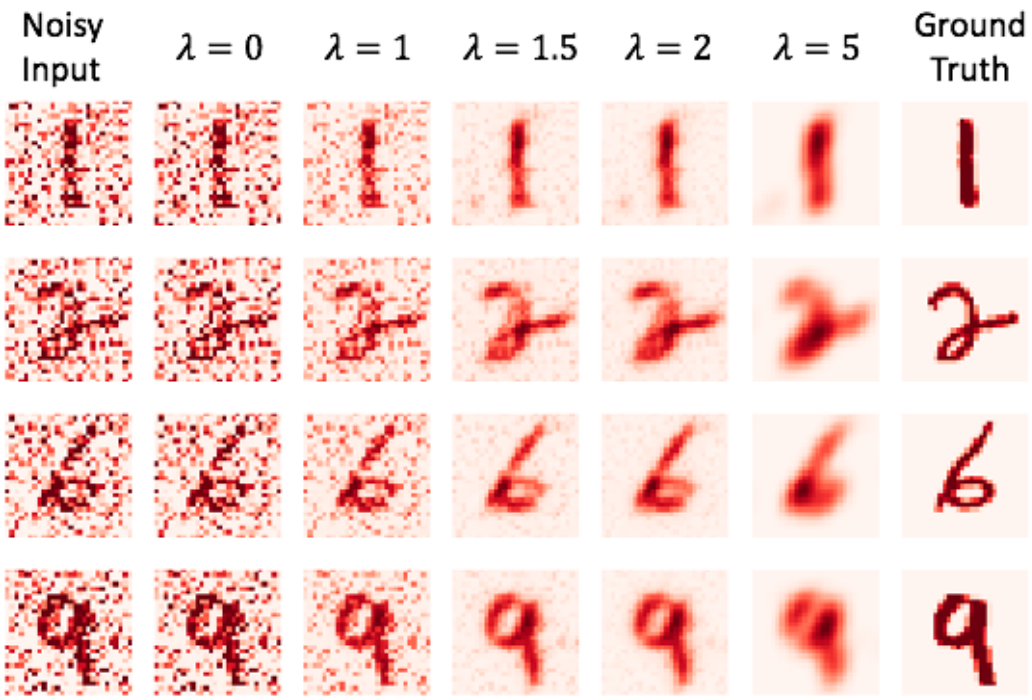


Figure 4.4: The output of the denoised results with different λ value. $\lambda = 1.5$ is the optimal in these samples

Table 4.1: Comparing the results obtained at different λ value. $\lambda = 1.5$ is the optimal in these samples

λ	MSE	Pearson Correlation	PSNR
Noisy Input	0.1156	0.5504	9.3
0.0	0.1090	0.5545	9.62
1.0	0.0487	0.7441	13.13
1.5	0.0252	0.8780	15.99
2.0	0.0262	0.8670	15.81
5.0	0.0428	0.7567	13.68

According to Eq.4.5, we expect that larger λ value will lead the model to remove noise more aggressively, and smaller λ will keep more information of the original noisy data. Herein, we investigate the impact of λ by setting λ with five different values $(0.0, 1.0, 1.5, 2.0, 5.0)$ with optimal network topology of CNNs. In Table 4.1, we compare the dissimilarity between denoised data and the original clean data quantitatively. The λ value between 1 and 2 obtain good results, and the outcome is the best when $\lambda = 1.5$ We draw the noisy input, original clean data and the denoised data with different λ value for several examples in Fig. 4.4. Consistent with the quantitative metrics, the best result is obtained when $\lambda = 1.5$. Clearly, when λ is small, the denoised output data still contains high-level noise, and the noise looks the same with the noise in the input data, indicating the model mostly "copied" the input data to the output without removing noise. On the other side, when λ is large ($\lambda = 5$), the noise is nearly completely removed, however, the signal is also partially removed.

4.3.4 Compare denoised data with averaging multiple experimental replicates

Obtaining multiple replicates need extra resources, and the previous discussion has proven our approach can achieve higher quality of the experimental data than a single replicate. Here, we compare the performance of our method with averaging results

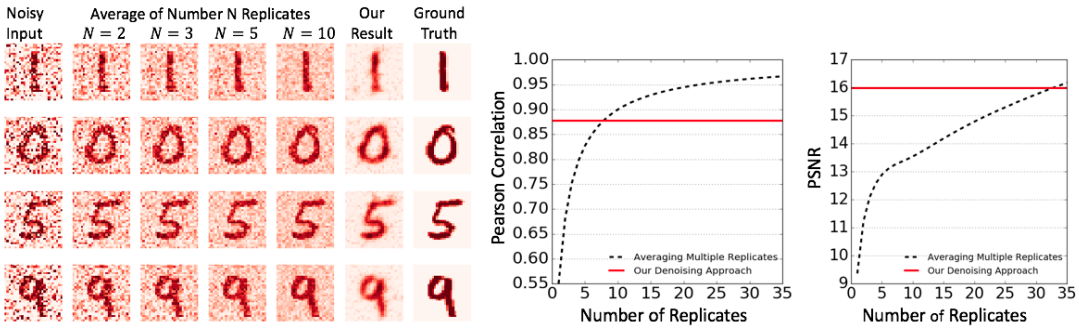


Figure 4.5: Compare our approach with averaging multiple experimental replicates. On the left, we plot noisy input data, our denoised result, ground truth and the result generated by averaging multiple noisy replicates. On the right, we evaluate the performance in Pearson Correlation and PSNR. Our denoised result is better than averaging the results of 5 and 30 parallel experiments in measurement of Pearson correlation and PSNR, respectively

from multiple replicates. To simulate multiple replicates of the experimental data, we employ the same procedure independently to add noise to the original clean data X to obtain multiple corrupted version of $\tilde{X}_i$. Then, we average the number of noisy data (N) to compare with the original clean data using the same metrics (Pearson correlation, PSNR). In Fig. 4.5, we give several examples as well as the quantitative metrics to compare the performance. In the right panel of Fig. 4.5, our denoised result is equivalent to the average of more than eight experimental replicates. From the examples shown in the left panel, our denoised result is at least better than average three replicates. By comparing the performance, we can conclude that by applying our denoised approach, at least 2/3 experiment resources can be saved to obtain the same quality of the data.

4.4 Application on Hi-C data

Denoising of Hi-C datasets is desired by biologists to enhance sign-to-noise ratio. Hi-C also provides a proper example for evaluating the performance of our new approach. In this work, to train the denoised network $F(\Theta)$ for Hi-C, we employ two biological replicates generated independently by the same protocol including benchwork as well

as data processing (Rao et al. 2014).

4.4.1 Preprocessing

The Hi-C signal intensity N_A, which reflects the distance in 3D space, is strongly effected by the distance between two loci along the DNA sequence (a.k.a genomic distance), and it is important to compare the *relative* signal intensity N_R with other pair loci on the same genomic distance. Therefore, we calculate the *relative* signal intensity N_R by Eq. 5.5

$$N_{R\ (i,j)} = \frac{N_{A\ (i,j)}}{M_A(|j-i|)} \tag{4.8}$$

where $N_{A\ (i,j)}$ is the Hi-C signal intensity between loci i and j, and $M_A(|j-i|)$ is the mean value of Hi-C signal intensity at genomic distance $|j-i|$

A Hi-C dataset is a large matrix spanning the entire genome, so we divide the whole Hi-C matrix into 40×40 patches with proper overlapping to cover the loss of surrounding region during the convolutional operation. Since most of the biologically meaningful features exist to the region when loci are close to each other, we pick the patches containing the region where two loci are less than 400 loci long.

4.4.2 Results

We employ the same topology of CNNs and use the same settings with the Siamese network on the simulated MNIST data. The training data is obtained from the first chromosome, and the testing data is obtained from Chromosome 18. After denoised on each patch of the testing data sets, we recombine the patches into the whole Hi-C matrix. To compare with Hi-C matrix in the absolute intensity, we also multiply the mean at its genomic distance to denoised Hi-C matrix to restore the *absolute* value.

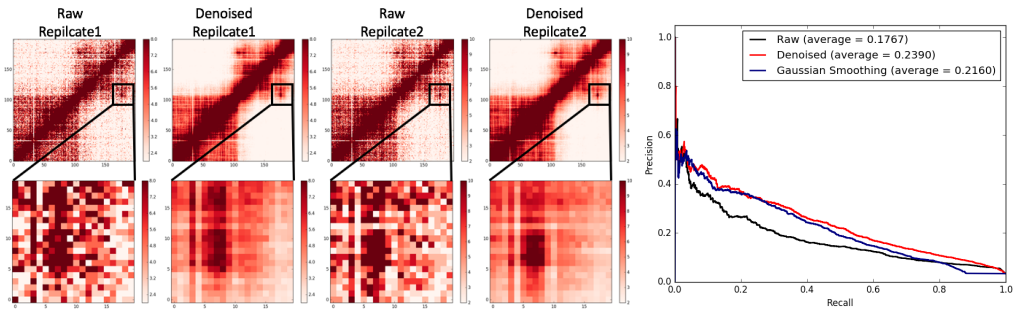
In Figure. 4.6a, we illustrate our denoised result by plotting the Hi-C interaction heatmap. Since there is no ground truth of the experiment, we list the raw and

denoised data of both replicates. From the comparison, we can observe that on the raw data, although two replicates share the major features (e.g. domain and loop peak), the detail is much different between two replicates because of the noise. The denoised process removes most of the noise and obtains very similar output from two different replicates.

Since the clean, true signal of Hi-C is unknown, we validate the model performance by another type of biological data set, CTCF ChIA-PET (Tang et al. 2015), of this type of cells. The CTCF ChIA-PET also reflects the 3D interaction between loci on the genome, but only focuses on the loci where the CTCF exists. CTCF is a protein closely related to the formation of the 3D structure of the genome, and the loci pair with CTCF ChIA-PET signal are expected to have stronger *relative* Hi-C signal as well. We pick all of the signal points which are eligible to have CTCF ChIA-PET signal, then define those which signal CTCF ChIA-PET observed as positive samples and those eligible but no signal observed as negative samples. After obtaining the positive and negative labels, we employ precision-recall curve to compare the performance of raw and denoised Hi-C matrix in Fig. 4.6b. As expected, our denoised Hi-C matrices outperform the raw matrix, and the best result is obtained when $\lambda = 1.5$, which is consistent with the simulation data.

4.5 Conclusions

In this work, we propose a novel approach to learn the denoising model at the situation where ground truth is unavailable as the training set. The denoising model is composed of convolutional neural networks which have been proven to be effective in noise removal. Our approach solves the challenge that no clean version of the training set is available in experimental science. We design the Siamese structure and proper loss function, allowing the model to learn to differentiate the true signal and the noise. The approach is tested on both simulated data and real experimental



(a) (b)

Figure 4.6: The performance of the denoising network on Hi-C data. 4.6a is an example focusing on the region with a loop peak (Rao et al. 2014), and denoised data enhances the signal of the peak by removing the noise in the background; 4.6b is the biological validation by CTCF ChIA-PET data (Tang et al. 2015) by the precision-recall curve, indicating the biological significance of our denoising approach.

data from Hi-C and is proven to be effective in noise reduction.

Our approach can be applied to many experimental fields, especially for those with the signal in 2D dimension. When applying this approach to different fields, it is also important to study the level of the noise removal (λ value) to balance the noise reduction and signal loss. Since the mechanism of this strategy is removal the volatile noise from the stable signal, it has great potential to several fields other than experimental data(e.g. remove isolated moving objects from stable landscape in image).

Chapter 5

Multiple-level Comparative Analysis of Hi-C Data by Triplet Network

5.1 Introduction

The Hi-C technique (Lieberman-Aiden et al. 2009; J. R. Dixon et al. 2012; Jin et al. 2013) can measure chromatin interaction intensities between any two loci on the chromosomes, and has become a powerful tool to dissect the spatial structure of the mammalian genomes. Since its birth, it has greatly expanded our knowledge of the 3D genome organization and has lead to several seminal discoveries such as *Topological Associating Domains*(*TADs*) (J. R. Dixon et al. 2012; Nora et al. 2012) and *chromatin loops* (Rao et al. 2014). As it attracts a lot of research interest in recent years, Hi-C has been performed in many cell and tissue types in several species and presents an invaluable resource for the study of gene regulation and in particular, how the changes in 3D genome organization can lead to differential cellular functions. Therefore, it is critical to develop the computational tool for quantitatively comparing Hi-C signals and identify the variations from different tissue/cell types.

5.1.1 Comparative analysis of Hi-C data sets

Currently, there are two major approaches for the comparisons of Hi-C interaction matrices (i.e., whole matrix-based approaches and feature-based approaches).

Hi-C data are usually presented as an $N \times N$ interaction matrix, where N is the number of bins along the genome and the value of each point indicate the number of pair-ended reads whose two ends are mapped to two different bins. Currently, Pearson correlation and Spearman correlation are commonly used to evaluate the similarities of the Hi-C matrices, and it works by converting the two-dimensional matrix into a one-dimensional vector and compute the coefficient (Lieberman-Aiden et al. 2009). More recently, a stratum-adjusted correlation coefficient method (HiCRep) (T. Yang et al. 2017) has been developed and has been shown to outperform Pearson and Spearman correlations. Essentially, it works by assigning the different weight for correlations at different distances.

However, the correlation-based approaches so far can only be employed to evaluate the reproducibilities/similarity for two Hi-C interaction matrices, and cannot reveal locations where the variations exist. Also, Pearson correlation and Spearman ranking correlation, as well as their transformation forms, cannot serve as a metrics of "distance".

Another potential class of matrix comparisons on Hi-C using the statistical models to detect the *significant interactions* on Hi-C (Ay, Bailey, and Noble 2014; Carty et al. 2017), and convert the Hi-C matrix into a binary form(i.e. *significant vs not significant*) for the comparative analysis. These approaches address the distance effect on Hi-C and remove some systematic bias in the Hi-C experiments but still have limitations: 1) the high-noise level on Hi-C can lead to falsely identified *significant interactions* sometimes unreliable, e.g. *significant interactions* didn't always overlap in two experimental replicates for the same cell type; 2) the statistical model usually assumes a particular kind of the distribution (e.g., Poisson (Ay, Bailey, and Noble 2014) and negative binomial (Carty et al. 2017)) of the Hi-C interaction intensity, and the assumption may be too simplified considering the complexities of the Hi-C data.

3) Matrix-wise approaches don't consider the surrounding region when calculating the probability of the significance.

Feature detection and comparison on Hi-C

Another approach to compare Hi-C data is to extract certain features from Hi-C interaction matrix first and then perform the comparative analysis. As mentioned above, several patterns have been discovered from the Hi-C interaction heatmap, such as *TADs* (J. R. Dixon et al. 2012) and *A/B compartment*, and it is useful to compare the difference of these features across cell types.

Currently, a rule-based model is usually implemented to determine the patterns on the entire genome, such as *TADs* (J. R. Dixon et al. 2012), *loop peaks* (Rao et al. 2014), and *frequently interacting regions* (*FIREs* (A. D. Schmitt et al. 2016)). Essentially, the feature extraction can also be regarded as the dimension reduction process, e.g. both (J. R. Dixon et al. 2012) and (A. D. Schmitt et al. 2016) employ the information of k upstream and k downstream interactions of a locus, and reduce the information from size $2 \times k + 1$ of to a single latent variable to represent state of the loci, and name the states as "*degree of upstream or downstream interaction bias*" and "*FIRE score*", respectively, based on the biological explanations. When assigning the states/score to a locus, the above-mentioned feature extractions incorporate the information from surrounding regions, leading to the several essential scientific findings on the chromatin spatial conformations. The feature detected by these approaches have been used as finding the variations (Rao et al. 2014; A. D. Schmitt et al. 2016; J. R. Dixon et al. 2012) on the Hi-C matrices but still have some disadvantages: 1) One rule can usually define one kind of state. Due to the complexity of Hi-C data, it is not practical to inspect all of the informative patterns on Hi-C by predefined rules; 2) Due to the noise level of the Hi-C, some features detected by rule-based approaches are not consistent with the results from another experimental replicate,

making it difficult to distinguish the true biological variations from random variations due to noises in the experiments.

5.1.2 Our contribution

Inspired by previous work on the Hi-C pattern extraction (J. R. Dixon et al. 2012; A. D. Schmitt et al. 2016), we project the high-dimension Hi-C interaction matrix to a vector of latent variables. Instead of pre-defining some rules for the projection, we employ the convolutional neural networks with trainable parameters to achieve this goal.

The convolutional neural networks contain multiple levels of the nonlinear filtering process, which enable the model to catch more complicated patterns comparing with simple arithmetic operations in the rule-based approaches. After obtaining the latent variables of the Hi-C matrices, the Euclidean distance is calculated to represent the distance between the Hi-C matrices.

The input to our triplet network are three Hi-C interaction matrices: two of them are biological replicates from the same cell type, and the third one is from a different cell type. We assume the difference between two biological replicates are minimal and are introduced by normal experimental variation. Using the triplet loss as the training target, the model can automatically update the parameters in the convolutional neural networks to obtain optimal latent variables that contain most significant patterns in the raw Hi-C data.

When comparing two Hi-C interaction matrices, it is challenging to distinguish the actual biological difference from the random noises. Here, our triplet network uses the two experimental replicates from the same cell types as an estimate of the natural noise level, and further employ this information to capture variations are between two cell types are above the random noise level.

Finally, we adopted the systematic occlusion approach in our framework, which

can identify specific variation regions on two Hi-C maps. The output of this step is a heatmap of *different scores* and its size is the same as the original Hi-C matrices. The *different scores* obtained by HiCComp not only contain the information of the corresponding pixels of raw Hi-C matrix but also considering the nearby regions as well as the noises generated in the Hi-C experiments. The entire process has no manual intervention, and all of the results are generated automatically from the model.

5.2 Method

5.2.1 Deep Neural Networks

In recent years, deep neural networks, also known as deep learning, achieve great success in multiple domains (Goodfellow, Bengio, and Courville 2016; LeCun, Bengio, and G. Hinton 2015; Schmidhuber 2015). In genomics and epigenomics, deep learning has been applied to predict protein binding sites of DNA (Alipanahi et al. 2015; Jian Zhou and Troyanskaya 2015; H. Zeng et al. 2016; Quang and X. Xie 2016) and the experimental data enhancement (Koh, Pierson, and Kundaje 2017; Y. Zhang et al. 2017). Deep learning employs multi-layered neural networks to process the raw input data multiple times and extracts relevant high-level features of the raw data. The parameters in the deep neural networks are updated by gradient-based optimization strategy, and the criterion for the optimization is loss function. The loss function is carefully designed to reflect the gap between network output and the ideal cases. Based on the training datasets provided and the loss function, the neural networks will extract the essential features in the raw data to minimize the loss in the training data sets.

5.2.2 Triplet Network

In the typical prediction task (including classifications and regressions) of deep learning, a large volume of labeled samples are required to train the network, where the labels reflect the real state of the samples. In the context of Hi-C data sets, we implemented the triplet network with corresponding triplet loss to train the network. Triplet loss (Hoffer and Ailon 2015; Balntas et al. 2016; J. Wang et al. 2014; Wohlhart and Lepetit 2015) is a ranking-based loss function and widely used to compare the similarity of images. The network topology in our model is shown as Fig. 5.1.

The input samples for triplet network are in form of (X, X^+, X^-), where X is *anchor*, X^+ is *positive* which belongs to the same class of X, and X^- is *negative* which belongs to different class of X. Specifically in this project, (X, X^+, X^-) are Hi-C matrices: *anchor* X and *positive* X^+ are Hi-C data from two biological replicates in the same cell type, and *negtaive* X^- is Hi-C data from another cell type.

The three instances in input samples X, X^+ and X^- are processed independently through the multilayer neural networks $F(X, \Theta)$ with identical topology and shared parameters for the feature extraction and representation, and converted to three vectors of latent variables Z, Z^+ and Z^-, respectively. The multi-layer neural networks $F(X, \Theta)$ for each input instance have shared parameters Θ, and Θ will be updated in the training process via gradient descent according to the triplet loss function.

After obtaining the latents variables Z, Z^+ and Z^-, the Euclidean distance D^+ and D^- are calculated, where $D^+ = ||Z^+ - Z||$ and $D^- = ||Z^- - Z||$. The triplet loss $L(D^+, D^-)$ is calculated according to the Eq. 5.1.

$$L(D^+, D^-) = \frac{1}{N}\left(\sum_{i=1}^{N} \max\{D^+ - D^- + M, 0\}\right) \tag{5.1}$$

where D^+ and D^- the Euclidean distance from *positive* and *negative* to *anchor*, respectively, and M is the margin value. The margin M prevents the model lazily set all latent variables to zero to achieve $L = 0$. In this project, we set $M = 1$ arbitrarily

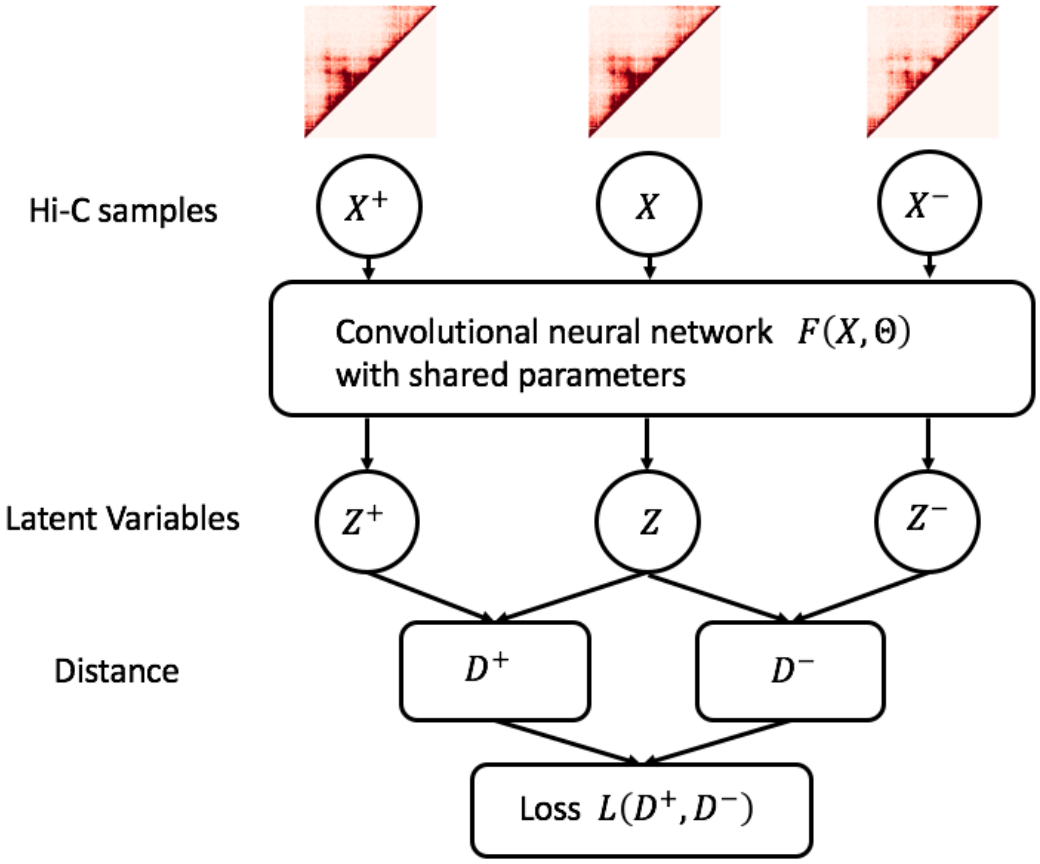


Figure 5.1: The network topology of our approach. The model proposed in this work contains three convolutional neural networks, with identical structures and the shared parameters. For each input Hi-C sub-matrix, the convolutional neural networks convert the Hi-C from its raw form to a vector of latent variables. Each sample contains three Hi-C sub-matrix from the same genomic location. Typically, *anchor* X and *positive* X^+ are from two biological replicates of the same cell types, and *negative* X^- is the Hi-C data in another cell type. All of the Hi-C sub-matrix shown in Fig. 5.1 are from chromosome 18: 6.15M-6.25M. The X and X^+ are from GM12878, and X^- is from K562. The Euclidean distances from *anchor* to *positive* $D^+$and *negative* D^- are calculated to pass to the loss calculation. The loss function is shown in Eq. 5.1.

since the range of absolute value of Z, Z^+, Z^- doesn't affect our the result, and we focus on the relative distance between D^+ and D^-.

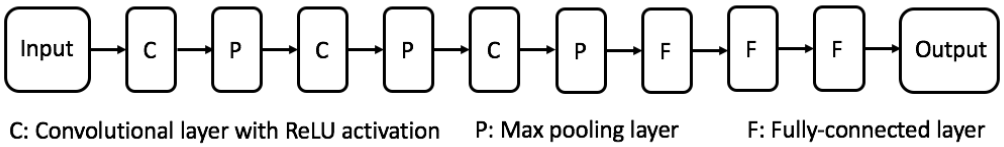


Figure 5.2: The structure of the convolutional neural networks. Our networks contain convolutional layers with ReLU activation (Glorot, Bordes, and Bengio 2011), max pooling layers, and fully-connected layers. In a typical structure, the networks include three convolutional layers, three max pooling layers, and two fully connected layers. The output is a vector of latent variables, and the length of the vector, as well as detail about the structure of convolutional neural networks, will be discussed in section 5.3.1.

5.2.3 Convolutional Neural Networks

In this work, $F(X, \Theta)$ is implemented by convolutional neural networks (LeCun et al. 1998), which is the combination of multiple convolutional layers, nonlinear activation layers, pooling layers, and fully connected layers. Convolutional neural networks are powerful to detect the highly correlated local motifs in the raw data, and such local motifs are invariant to the locations. In Hi-C interaction heatmap, the chromatin interaction patterns, such as loop peaks or boundaries of TADs, can be regards as such local motifs. The structure of the networks in $F(X, \Theta)$ is shown in Fig. 5.2, and we will discuss the detail hyper-parameters later. We employ standard gradient descent (SGD) (Sutskever et al. 2013) and L2 regularization in the training to update parameters Θ in $F(X, \Theta)$ to minimize triplet loss $L(D^+, D^-)$.

5.2.4 Sample Processing

Unless otherwise noted, all of the Hi-C data used in this project are raw Hi-C interaction matrices from (Rao et al. 2014) at 10kb resolution. We used the two biologically replicates in GM12878 as *anchor* and *positive*. We chose IMR90 and K562

as the source of *negative* for training and testing, respectively. To ensure that all of the Hi-C matrices have the same overall interaction intensities, we down-sampled all of the Hi-C interaction matrices to the same sequencing depth based on the lowest sequenced replicate (K562 in this case).

Considering most of significant chromatin interactions on Hi-C matrix are within $1Mb$ TADs, we divided the Hi-C matrix into 100×100 segment along the diagonals (Fig. 5.3) to generate the data sets (X_1, X_1^+, X_1^-), (X_2, X_2^+, X_2^-), $...(X_i, X_i^+, X_i^-)$,... (X_n, X_n^+, X_n^-). The samples are divided in the same order to ensure the samples with the same index i present the Hi-C patches from the same genomic locations for all kinds of cell types.

In the training dataset, there is no overlap between the samples, and testing data sets may contain overlapped samples depending on the usage. The lower right part of the Hi-C is set to zero for all of the samples since the Hi-C patches are symmetrical along the diagonal. The training data sets utilized *GM12878 rep. 1*, *GM12878 rep. 2*, and *IMR90* as the *anchor*, *positive* and *negative*, respectively. Chromosomes 1-8 are used as training set, and 6218 samples are generated.

5.3 Results

5.3.1 Impact of choice of hyperparameters

Number of latent variables

As discussed above, feature extraction is essentially a process of the dimension reduction for the raw data. In the previous work, one latent variable are used to represent the *chromatin interaction bias* (*directionality index*) (J. R. Dixon et al. 2012) and *FIREs score*) (A. D. Schmitt et al. 2016)), and N latent variables are used to represent a Hi-C matrix with length of N.

In this work, suppose each Hi-C submatrix can be represented by n variables,

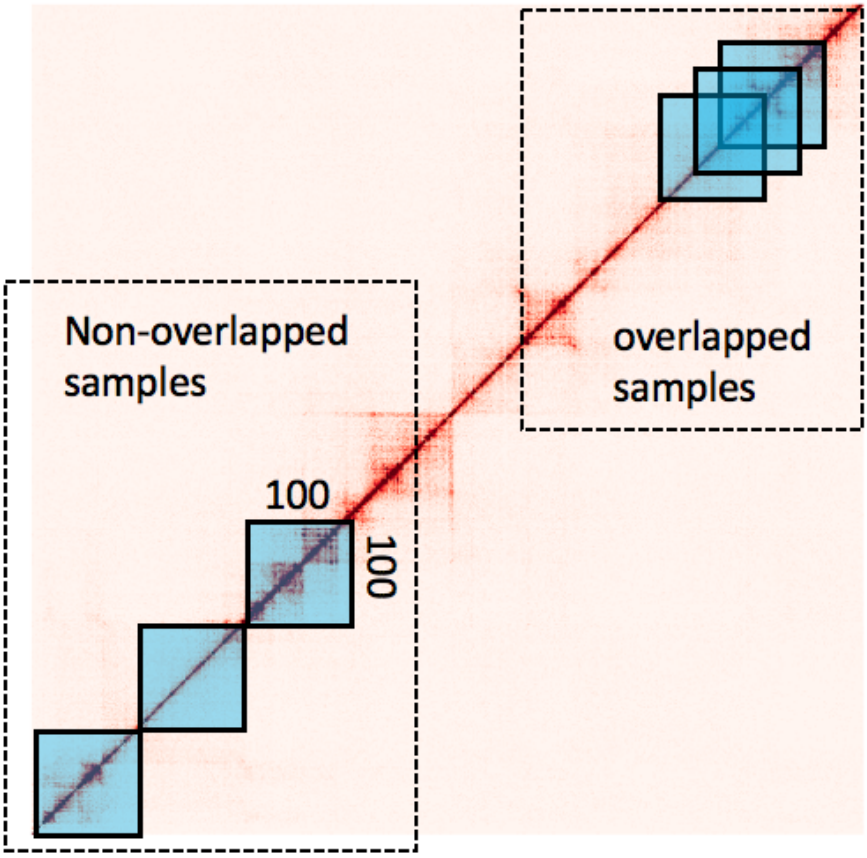


Figure 5.3: Dividing chromosome-wide Hi-C matrix into sub-matrix for training and prediction. The Hi-C matrix is divided into small 100×100 sub-matrices along the diagonal. On the lower left, we show the dividing without overlap for the training and validation data sets. On the upper right, the sub-matrices are overlapped with each other for the downstream analysis

the dimension of entire Hi-C matrix is reduced to a $n \times N$ when we divided the Hi-C matrix to sub-matrices of $(1, 100), (2, 101), ...(N - 100, N)$, when the size of the submatrix is 100×100. Since Hi-C is detecting the 3D structure of the chromatin, the number of the latent variables should be at least 3 ($n \geqslant 3$), and $n = 3$ if chromatin is static and the entire group of the cells in the experiment share the same chromatin 3D structure. Therefore, we set up our experiments starting from $n = 3$ and with an increment of 2. The number of latent variables is adjusted by the dimension of the output in the last fully-connect layer with other hyperparameters in the model

remain the same.

We trained the model on *(GM12878 rep. 1, GM12878 rep. 2, IMR90)* and calculate the validation error on the same chromosomes of *(GM12878 rep1, GM12878 rep2, IMR90)* . As shown in Fig. 5.4, the triplet loss decrease as number of latent increase from $n = 3$ to $n = 7$, and remain the same when $n = 7$. Therefore, we use 7 latents variables to represent the 100 Hi-C patches.

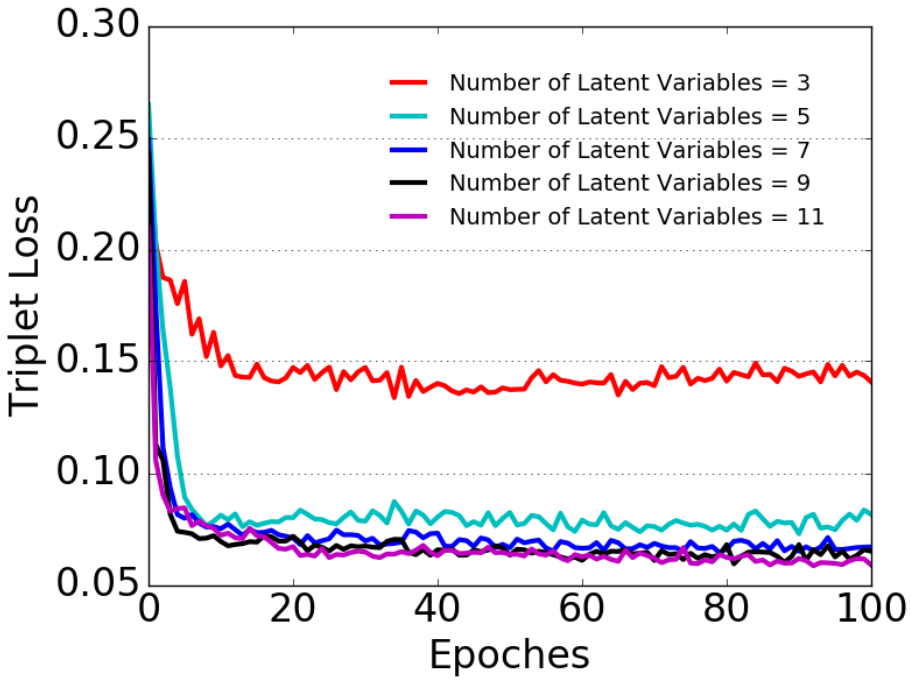


Figure 5.4: Relationship between the loss and the number of latent variables. We tested the different length of latent variables Z by changing the output of last fully-connected layer with other hyper-parameters in the network unchanged.

Hyper-parameters in convolutional neural networks

As shown in Fig. 5.2, the convolutional neural networks in our model contain several convolutional layers and pooling layers, so it is important to find the optimal number layers for the following study. The filter numbers for all of the convolutional layers are 8, and each 2D convolutional layer is followed by a 2D max pooling layer with the size of 2×2. After the convolutional layers, there are two fully connected layers

with the number of hidden nodes 500 and 10. In the comparison, we implemented the four networks with 1 to 4 convolutional and pooling layers, respectively, and all of the settings remain constant except the number of layers. The 2D filters size of each layers are *9, 9-3, 9-3-3, and 9-3-3-3* for the models with 1, 2, 3, and 4 convolutional layers, respectively. As shown in Fig. 5.5, the model with 3 convolutional layers achieves the minimum triplet loss on validation dataset.

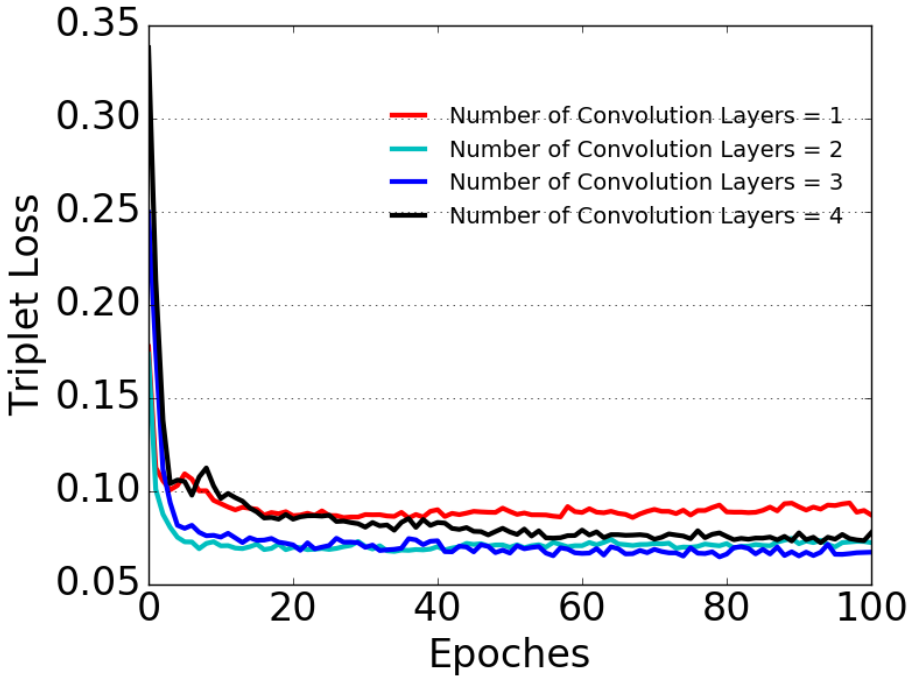


Figure 5.5: Performance of the convolutional neural networks with different convolutional layers.

5.3.2 Our proposed distance reflects the similarity of the chromatin state

We trained our convolutional neural networks with Hi-C interaction matrices from chromosomes 1-8 and tested its performance on chromosome 18. To generate the "*difference*" interaction matrix between GM12878 and K562, we divided chromosome 18 into 100x100 sub-matrix along diagonal with $100-1=99$ overlapped loci between adjacent samples. Donating the total bins on chromosome 18 is N, the sub-matrices with index 0 to $N-100$ are come from the location $(1,100),(2,101)...(N-100,N)$.

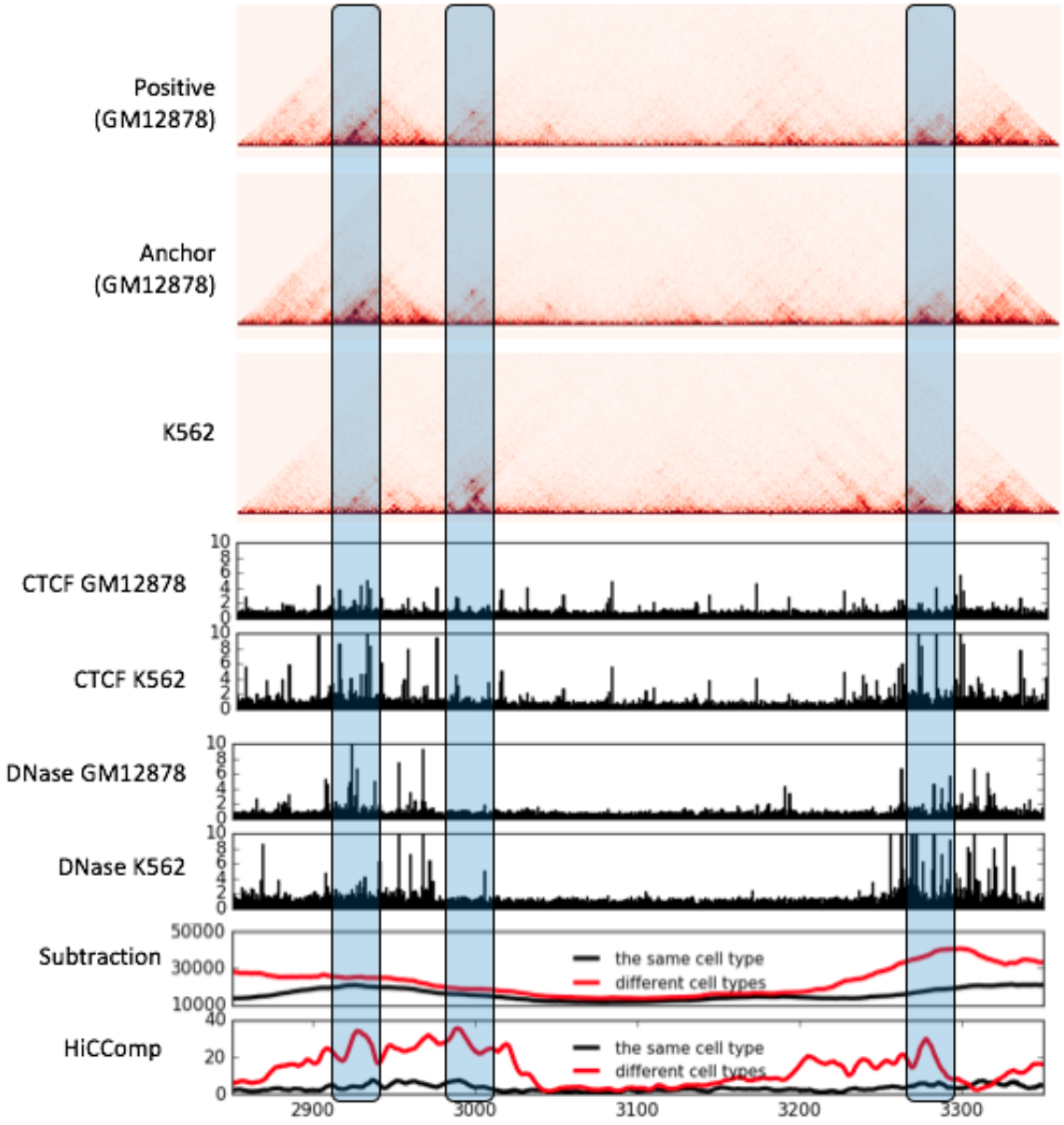


Figure 5.6: HiCComp identified variations in Hi-C interaction matrices. On the top, we plot the Hi-C interaction heatmap along the diagonal of K562 and two experimental replicates from the experiment. All of the Hi-C matrices are down-sampled to the same sequencing depth, and the color scale is the same ($min = 0, max = 50$). The middle part is the enrichment of the 1D epigenomic signal, and we show two ChIP-Seq tracks (CTCF and DNase) which have been shown to be related to the spatial structure. The lower part is the similarities evaluated by different approaches, and the similar on loci k reflects the sample range from $k - N/2$ to $k + N/2$. Besides our approach, we also implemented pixel subtraction of two 100×100 Hi-C matrix(Eq. 5.2).

For the initial evaluation, we use the central location to represent the similarity of the each sample, e.g., the sample (0:100) donates the similarity of loci 100/2=50. As for the baseline, we used a naïve approach where we subtract the matrix in GM12878 from the K562 interaction matrix.

$$D_{pixel} = \sum_{i,j} |a_{ij} - b_{ij}| \tag{5.2}$$

In Fig. 5.6, we draw a comparison of Hi-C matrix, 1D epigenomic signals related to the spatial structure (CTCF (Rao et al. 2014; J. R. Dixon et al. 2012) and DNase[(Schreiber et al. 2017; Mourad and Cuvier 2015; Sutskever et al. 2013)), as well as distance (D^+, D^-). Pixel subtraction and HiCComp can catch the general trend of the variations on the Hi-C. We also observed pixel-wise subtraction are too smooth and not very sensitive to small-scale variations, such as loops. It also looks the absolute interaction intensity strongly affects the distance calculated by pixel subtraction approaches. HiCComp identifies the exact locations of the Hi-C variations across cell types.

5.3.3 HiCComp identifies the exact locations of the Hi-C variations across cell types

By initial visual inspection, our proposed feature-level distance (D) can reflect the difference among Hi-C matrices, but such dissimilarities are based on the average of 100×100 Hi-C sub-matrix. Considering the region with $100 \times 10kb = 1mb$ base pairs is still too large to identify biologically significant variations, we need to further identify the precise locations of the variations between cell types.

In computer vision, systematic occluding the input images and record the change in the output is an efficient way to detect the essential regions for making the accurate predictions. For example, in dog breed classification, occluding dog's face will make the prediction accuracy significantly drop while occluding another part will not effect

the output much (Zeiler and Fergus 2014). Therefore, we perform the occluding for our Hi-C samples to identify the critical regions for the model to distinguish whether the Hi-C patches are from the same cell types or not.

Impact of the occlusion on the samples

In Fig. 5.7 and Table 5.1, we demonstrate how the change in the distance D^+ and D^- when occluding different 8×8 windows on the Hi-C sample. For all three Hi-C sub-matrices in the same sample set, the occlusion windows located in the same genomic location, and we fill the windows with zeros to replace the original Hi-C interaction intensities. As shown in Fig. 5.7 and Table 5.1 when we the occluding window at the location where the Hi-C patch are highly similar (Occlusion 1), the gap between D^+ and D^- is nearly unchanged. However, when the occlusion window is located in a region where the chromatin interactions are different, it leads to a dramatic change in the difference score (Occlusion 2). For example, in Fig. 5.7 lower panel, the occlusion window is located within a strong TAD region in K562, but there is no signal in GM12878, and as a result, it deviated dramatically from the original matrix.

Table 5.1: Difference in the change of the distances D caused by different occlusions as shown in Fig. 5.7

	D^+	D^-	$max(0, D^- - D^+)$	change
No occluding	4.16	5.46	1.30	NA
Occluding 1	4.07	4.31	0.24	1.06
Occluding 2	4.29	5.59	1.30	0

Generate the difference scores using systematic occlusions in the entire Hi-C matrix

Since the occlusion can effectively reflect the importance of a region, we systematically occluded the entire Hi-C sample to obtain the variation scores on entire Hi-C heatmap

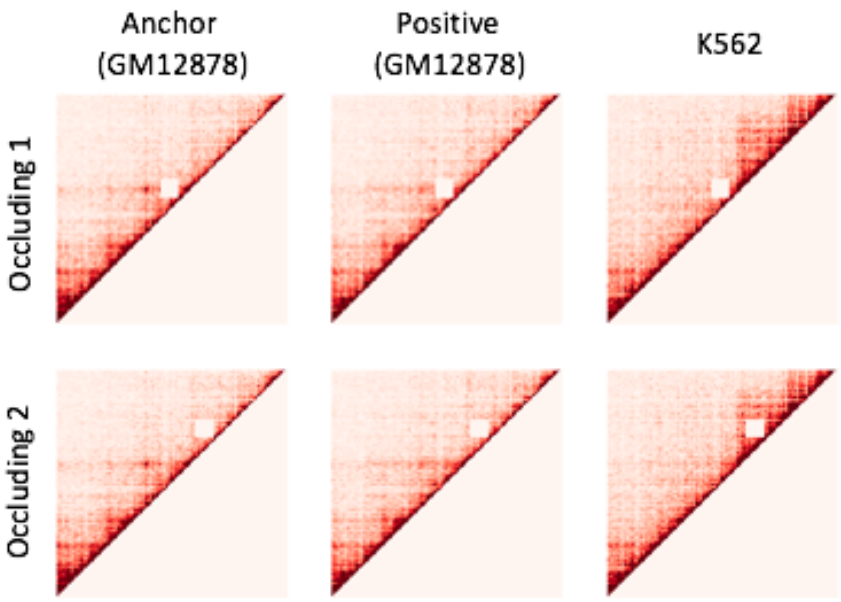


Figure 5.7: Occluding different part of the samples has difference effect in determining the difference between two Hi-C matrices. Occlusion 1 removes the region all sample Hi-C matrices are highly similar, and Occlusion 2 removes the regions where the interaction pattern is different. The impact of the quantitative metrics is shown in Table. 5.1

as the following steps.

(a) For each sample (X, X^+, X^-) in the testing data sets, we slid the occlusion window step by step through the entire matrix for three Hi-C sub-matrices. Donating the sliding window size is $w \times w$ and the Hi-C matrices size are $N \times N$, this step will generate $(N-w) \times (N-w)$ new occlusion samples. In this study, we set window size w to 8 considering the known pattern such as loops and domain boundaries are usually smaller than 8 in at least one dimension.

(b) Run the prediction on occlusion samples using the model as testing samples, and calculate the new gap G_i from the new distance according to Eq. 5.3, then calculate the gap change during the occlusion by Eq. 5.4 as the *difference scores* the pixels inside the occlusion window.

$$G_i = D_i^- - D_i^+ \tag{5.3}$$

where i is the index in the newly generated occlusion samples.

$$S_i = max(0, G_0 - G_i) \tag{5.4}$$

where G_0 is the gap for the original unoccluded sample and also calculated by Eq. 5.3. The *difference scores* can be explained as the gap shrunk during the occlusion process. If the difference of D^+ and D^- decrease a lot after the region is occluded, the area is important for the model to distinguish whether the sample pairs are from the same cell types. Otherwise, the model cannot infer the variation across cell types at this region is beyond the experimental variations between two experimental replicates.

(c) A pixel on Hi-C patches may be covered by occlusion windows many times in different occlusion samples, and we calculate the *difference scores* for this pixel by averaging the *difference scores* of all occlusion samples containing this pixel. After generating the *difference scores* matrix for each sub-matrices, we combined all of the samples to generate the *difference scores* for the entire chromosome. For the regions covered in several samples, similar to the previous step, we use the mean of all scores in the pixel as the final *idifference scores*.

For *difference scores* matrix from pixel subtraction, we also divided the Hi-C interaction value C_i on by the mean of the value in the same distance to eliminate the distance effect on the Hi-C matrix.

$$C_{(x,y)} = \frac{C^{raw}_{(x,y)}}{C_{mean}(|y-x|)} \tag{5.5}$$

where $M_{mean}(|y-x|$ is the mean value of C^{raw} at genomic distance $|y-x|$

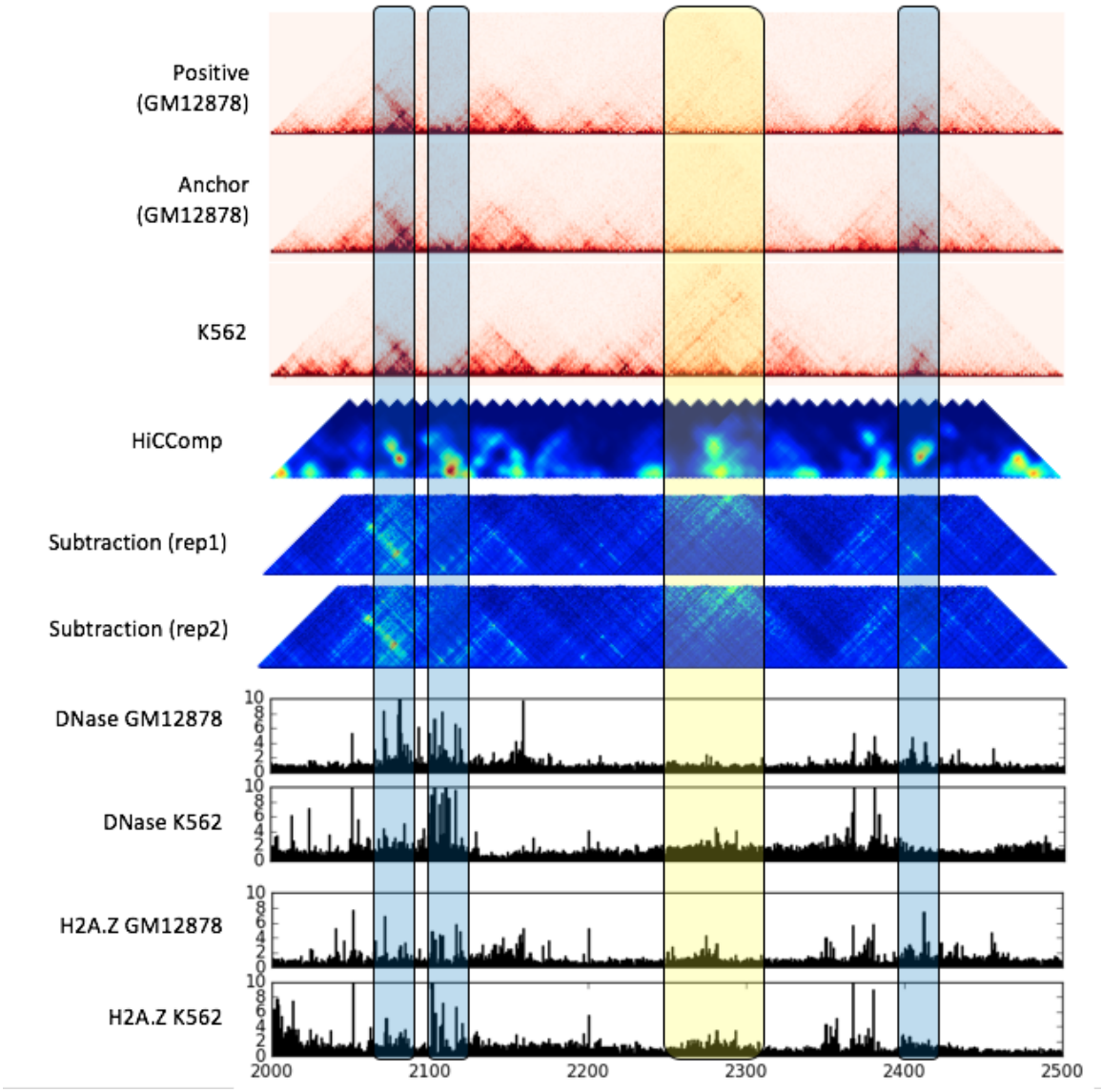


Figure 5.8: Systematic occlusion helps HiCComp capture dynamic interactions in Hi-C interaction matrices. Upper: Hi-C raw matrix(color scale: 0-50); Middle: the *difference scores* matrix generated by HiCComp and other approaches(warm color indicates high *difference scores*); Lower: enrichments of the epigenomic markers(the height indicates the relative enrichment level comparing with the global average). The location of the variations called by our approach, HiCComp, are strongly linked with the enrichment of the 1D epigenomic markers.

Systematic occlusions reveal key interaction regions.

The result from the systemic occluding is shown in Fig. 5.8. The top part contains Hi-C heatmaps from GM12878 and K562, and the *difference score* maps are shown in the middle. For the comparison purpose, we also plot the enrichment level of DNase and H2A.Z for both cell types in the lower panel. Surprisingly, our model clearly indicates several significant variations which are not very obvious by visual inspection (marked by blue boxes). Through the comparison with the 1D epigenomic signal, these variations are linked with the different chromatin state. The *difference score* from pixel subtraction also catch some variations but miss several important regions. It is interesting that some significant area can be observed manually (yellow shading region) is indicated by HiCComp as the moderate level of the variations, and the enrichment of DNase and H2A.Z are also only slightly different across two cell types in this region.

Validate the proposed Hi-C variations by the enrichment of 1D epigenomic markers

The 2D signals of Hi-C are strongly linked with the 1D epigenomic makers (J. R. Dixon et al. 2012; Rao et al. 2014; Sanborn et al. 2015; J. W. Ho et al. 2014; Y. Zhu et al. 2016; Pancaldi et al. 2016; J. Huang et al. 2015; Y. Chen et al. 2016), which are usually generated by Chip-seq. To quantitatively evaluate our result, we calculate the enrichment of the epigenomic marker(obtained from (Bernstein et al. 2010)) using the same bin size as Hi-C(10kb), then define the difference of the enrichment by Eq. 5.6.

$$D_{Enrich,i} = \frac{|Enrich_{1,i} - Enrich_{2,i}|}{Enrich_{1,i} + Enrich_{2,i} + tiny} \tag{5.6}$$

where i is the location of the bin, $Enrich_{1,i}$ and $Enrich_{2,i}$ are the enrichment level on bin i for two cell types, and $tiny$ is a tiny number to avoid dividing by zero. The range of $D_{Enrich,i}$ is 0 to 1.

The enrichment difference is shown in Fig. 5.9 for chromosome 18 as the boxplot. The baseline (as shown in black) is the $D_{Enrich,i}$ on all bins of chromosome 18 ($\sim$ 7700 sample). To calculate the difference of 1D epigenomic signal on the locations where Hi-C variations occur, we first take top 1000 bins with highest *difference scores* for all of the methods (HiCComp, Pearson correlation, and pixel subtraction). For each selected bin $B_{i,j}$ from the 2D matrix of *difference scores*, we use the difference of the enrichment on the corresponding 1D bins $D_{Enrich,i}$ and $D_{Enrich,j}$ to represent its location in the 1D chromosome. For the 1D bins which are included in several selected 2D bins, we just include all of them without removing duplicate, so we obtain a vector containing 2000 D_{Enrich} for the box plot for the three Hi-C variation detection approach. As shown in Fig. 5.9, the regions detected by HiCComp have significant higher *difference scores* on the multiple types of the epigenomics marker, especially for the CTCF, DNase, H1A.Z, H2K4me1. The Hi-C variations called by pixel subtraction also have the higher level in the difference on the 1D epigenomics marker but not as significantly as HiCComp.

5.3.4 Conclusion

Here we present HiCComp, a comprehensive framework based on convolutional network and systematic occlusions, for the comparative analysis of Hi-C from different tissue/cell types. HiCComp can quantitatively compute the differences of Hi-C data at whole chromosome-level and also simultaneously identify the particular locations of the variations. We observe that the variations in Hi-C data identified by HiCComp are enriched for transcription factor binding sites and histone modifications that are associated with cis-regulatory functions, suggesting these variations in 3D genome structure are potentially gene regulatory events. In this work, we have demonstrated the usage of our framework to distinguish and identify variations in Hi-C data from two cell types. In the future, we will further expand this work so that it can process

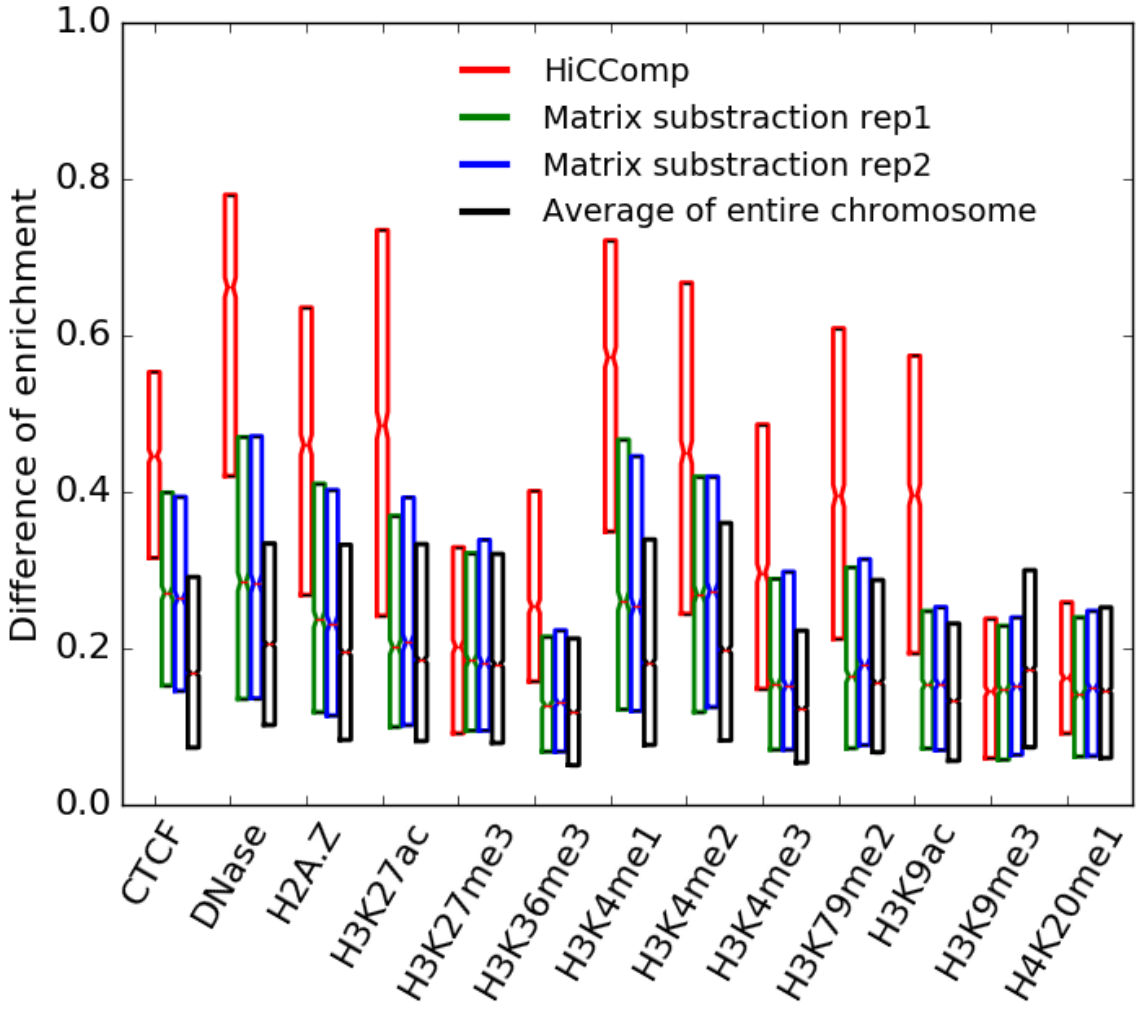


Figure 5.9: The variations of the Hi-C can be validated by the enrichments of the epigenomic markers. The box plot shows the 25th percentile, mean, and 75th percentile of each sample group. The groups shown in color contain the locations of the variations on Hi-C called by different approaches, and the baseline of the entire chromosome is shown in black.

multiple cell types together.

Bibliography

Alipanahi, Babak et al. (2015). "Predicting the sequence specificities of DNA-and RNA-binding proteins by deep learning". In: *Nature biotechnology* 33.8, pp. 831–838.

Angermueller, Christof et al. (2016). "Deep learning for computational biology". In: *Molecular systems biology* 12.7, p. 878.

Angermueller, Christof et al. (2017). "Accurate prediction of single-cell DNA methylation states using deep learning". In: *bioRxiv*, p. 055715.

Ay, Ferhat, Timothy L Bailey, and William Stafford Noble (2014). "Statistical confidence estimation for Hi-C data reveals regulatory chromatin contacts". In: *Genome research* 24.6, pp. 999–1011.

Balntas, Vassileios et al. (2016). "Learning local feature descriptors with triplets and shallow convolutional neural networks." In: *BMVC*. Vol. 1. 2, p. 3.

Baù, Davide and Marc A Marti-Renom (2011). "Structure determination of genomic domains by satisfaction of spatial restraints". In: *Chromosome research* 19.1, pp. 25–35.

Bayley, Martin J et al. (1998). "GENFOLD: A genetic algorithm for folding protein structures using NMR restraints". In: *Protein Science* 7.2, pp. 491–499.

Bengio, Yoshua, Yann LeCun, et al. (2007). "Scaling learning algorithms towards AI". In: *Large-scale kernel machines* 34.5, pp. 1–41.

Bernstein, Bradley E et al. (2010). "The NIH roadmap epigenomics mapping consortium". In: *Nature biotechnology* 28.10, pp. 1045–1048.

Bianco, Simone (2017). "Large age-gap face verification by feature injection in deep networks". In: *Pattern Recognition Letters* 90, pp. 36–42.

Bickmore, Wendy A (2013). "The spatial organization of the human genome". In: *Annual review of genomics and human genetics* 14, pp. 67–84.

Bonev, Boyan and Giacomo Cavalli (2016). "Organization and function of the 3D genome". In: *Nature Reviews Genetics* 17.11, pp. 661–678.

Bowie, James U and David Eisenberg (1994). "An evolutionary approach to folding small alpha-helical proteins that uses sequence information and an empirical guiding fitness function". In: *Proceedings of the National Academy of Sciences* 91.10, pp. 4436–4440.

Bromley, Jane et al. (1994). "Signature Verification using a "Siamese" Time Delay Neural Network". In: *Advances in Neural Information Processing Systems 6*. Ed. by J. D. Cowan, G. Tesauro, and J. Alspector. Morgan-Kaufmann, pp. 737–744. URL: `http://papers.nips.cc/paper/769-signature-verification-using-a-siamese-time-delay-neural-network.pdf`.

Burger, Harold C, Christian J Schuler, and Stefan Harmeling (2012). "Image denoising: Can plain neural networks compete with BM3D?" In: *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*. IEEE, pp. 2392–2399.

Carty, Mark et al. (2017). "An integrated model for detecting significant chromatin interactions from high-resolution Hi-C data". In: *Nature Communications* 8, p. 15454.

Chen, Yong et al. (2016). "De novo deciphering three-dimensional chromatin interaction and topological domains by wavelet transformation of epigenetic profiles". In: *Nucleic acids research* 44.11, e106–e106.

Chopra, Sumit, Raia Hadsell, and Yann LeCun (2005). "Learning a similarity metric discriminatively, with application to face verification". In: *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*. Vol. 1. IEEE, pp. 539–546.

Chua, Alvin L-S et al. (2010). "A genetic algorithm for predicting the structures of interfaces in multicomponent systems". In: *Nature materials* 9.5, pp. 418–422.

Consortium, ENCODE Project et al. (2012). "An integrated encyclopedia of DNA elements in the human genome". In: *Nature* 489.7414, p. 57.

Cremer, Thomas and Christoph Cremer (2001). "Chromosome territories, nuclear architecture and gene regulation in mammalian cells". In: *Nature reviews genetics* 2.4, pp. 292–301.

Dahl, George E et al. (2012). "Context-dependent pre-trained deep neural networks for large-vocabulary speech recognition". In: *IEEE Transactions on audio, speech, and language processing* 20.1, pp. 30–42.

Davis, Lawrence (1991). "Handbook of genetic algorithms". In:

Deaven, DM and KM Ho (1995). "Molecular geometry optimization with a genetic algorithm". In: *Physical review letters* 75.2, p. 288.

Dixon, Jesse R et al. (2012). "Topological domains in mammalian genomes identified by analysis of chromatin interactions". In: *Nature* 485.7398, pp. 376–380.

Dixon, Jesse R et al. (2015). "Chromatin architecture reorganization during stem cell differentiation". In: *Nature* 518.7539, pp. 331–336.

Dong, Chao et al. (2014). "Learning a deep convolutional network for image super-resolution". In: *European Conference on Computer Vision*. Springer, pp. 184–199.

— (2016). "Image super-resolution using deep convolutional networks". In: *IEEE transactions on pattern analysis and machine intelligence* 38.2, pp. 295–307.

Duan, Zhijun et al. (2010). "A three-dimensional model of the yeast genome". In: *Nature* 465.7296, pp. 363–367.

Eigen, David, Dilip Krishnan, and Rob Fergus (2013). "Restoring an image taken through a window covered with dirt or rain". In: *Proceedings of the IEEE International Conference on Computer Vision*, pp. 633–640.

Ernst, Jason and Manolis Kellis (2012). "ChromHMM: automating chromatin-state discovery and characterization". In: *Nature methods* 9.3, pp. 215–216.

Farabet, Clement et al. (2013). "Learning hierarchical features for scene labeling". In: *IEEE transactions on pattern analysis and machine intelligence* 35.8, pp. 1915–1929.

Farabet, Clément et al. (2012). "Scene parsing with multiscale feature learning, purity trees, and optimal covers". In: *arXiv preprint arXiv:1202.2160*.

Fraser, James et al. (2015). "Hierarchical folding and reorganization of chromosomes are linked to transcriptional changes in cellular differentiation". In: *Molecular systems biology* 11.12, p. 852.

Fraser, Peter and Wendy Bickmore (2007). "Nuclear organization of the genome and the potential for gene regulation". In: *Nature* 447.7143, pp. 413–417.

Fukushima, Kunihiko (1980). "Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position". In: *Biological cybernetics* 36.4, pp. 193–202.

Fullwood, Melissa J et al. (2009). "An oestrogen-receptor-&agr;-bound human chromatin interactome". In: *Nature* 462.7269, pp. 58–64.

Gardiner, Eleanor J, Peter Willett, and Peter J Artymiuk (2001). "Protein docking using a genetic algorithm". In: *Proteins: Structure, Function, and Bioinformatics* 44.1, pp. 44–56.

Glasner, Daniel, Shai Bagon, and Michal Irani (2009). "Super-resolution from a single image". In: *Computer Vision, 2009 IEEE 12th International Conference on*. IEEE, pp. 349–356.

Glorot, Xavier, Antoine Bordes, and Yoshua Bengio (2011). "Deep sparse rectifier neural networks". In: *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, pp. 315–323.

Golberg, David E (1989). "Genetic algorithms in search, optimization, and machine learning". In: *Addion wesley* 1989.

Goodfellow, Ian, Yoshua Bengio, and Aaron Courville (2016). *Deep Learning*. `http://www.deeplearningbook.org`. MIT Press.

Graves, Alex, Abdel-rahman Mohamed, and Geoffrey Hinton (2013). "Speech recognition with deep recurrent neural networks". In: *Acoustics, speech and signal processing (icassp), 2013 ieee international conference on*. IEEE, pp. 6645–6649.

Hinton, Geoffrey E, Simon Osindero, and Yee-Whye Teh (2006). "A fast learning algorithm for deep belief nets". In: *Neural computation* 18.7, pp. 1527–1554.

Ho, Joshua WK et al. (2014). "Comparative analysis of metazoan chromatin organization". In: *Nature* 512.7515, p. 449.

Hoffer, Elad and Nir Ailon (2015). "Deep metric learning using triplet network". In: *International Workshop on Similarity-Based Pattern Recognition*. Springer, pp. 84–92.

Hu, Ming et al. (2012). "HiCNorm: removing biases in Hi-C data via Poisson regression". In: *Bioinformatics* 28.23, pp. 3131–3133.

Hu, Ming et al. (2013). "Bayesian inference of spatial organizations of chromosomes". In: *PLoS Comput Biol* 9.1, e1002893.

Huang, Jialiang et al. (2015). "Predicting chromatin organization using histone marks". In: *Genome biology* 16.1, p. 162.

Jin, Fulai et al. (2013). "A high-resolution map of the three-dimensional chromatin interactome in human cells". In: *Nature* 503.7475, pp. 290–294.

Johnson, David S et al. (2007). "Genome-wide mapping of in vivo protein-DNA interactions". In: *Science* 316.5830, pp. 1497–1502.

Jones, Eric, Travis Oliphant, Pearu Peterson, et al. (2001–). *SciPy: Open source scientific tools for Python*. [Online; accessed <today>]. URL: `http://www.scipy.org/`.

Kamper, Herman, Weiran Wang, and Karen Livescu (2016). "Deep convolutional acoustic word embeddings using word-pair side information". In: *Acoustics, Speech and Signal Processing (ICASSP), 2016 IEEE International Conference on*. IEEE, pp. 4950–4954.

Kantz, Holger et al. (1993). "Nonlinear noise reduction: A case study on experimental data". In: *Physical Review E* 48.2, p. 1529.

Kelley, David R, Jasper Snoek, and John L Rinn (2016). "Basset: learning the regulatory code of the accessible genome with deep convolutional neural networks". In: *Genome research* 26.7, pp. 990–999.

Khalil-Hani, Mohamed and Liew Shan Sung (2014). "A convolutional neural network approach for face verification". In: *High Performance Computing & Simulation (HPCS), 2014 International Conference on*. IEEE, pp. 707–714.

Koh, Pang Wei, Emma Pierson, and Anshul Kundaje (2017). "Denoising genome-wide histone ChIP-seq with convolutional neural networks". In: *Bioinformatics* 33.14, pp. i225–i233.

Krizhevsky, Alex, Ilya Sutskever, and Geoffrey E Hinton (2012). "Imagenet classification with deep convolutional neural networks". In: *Advances in neural information processing systems*, pp. 1097–1105.

LeCun, Yann, Yoshua Bengio, and Geoffrey Hinton (2015). "Deep learning". In: *Nature* 521.7553, pp. 436–444.

LeCun, Yann et al. (1998). "Gradient-based learning applied to document recognition". In: *Proceedings of the IEEE* 86.11, pp. 2278–2324.

Lesne, Annick et al. (2014). "3D genome reconstruction from chromosomal contacts". In: *Nature methods* 11.11, pp. 1141–1143.

Leung, Danny et al. (2015). "Integrative analysis of haplotype-resolved epigenomes across human tissues". In: *Nature* 518.7539, p. 350.

Li, Heng and Richard Durbin (2009). "Fast and accurate short read alignment with Burrows–Wheeler transform". In: *Bioinformatics* 25.14, pp. 1754–1760.

Lieberman-Aiden, Erez et al. (2009). "Comprehensive mapping of long-range interactions reveals folding principles of the human genome". In: *Science* 326.5950, pp. 289–293.

Liu, Feng et al. (2016). "PEDLA: predicting enhancers with a deep learning-based algorithmic framework". In: *Scientific reports* 6, p. 28517.

Maas, Andrew et al. (2012). "Recurrent Neural Networks for Noise Reduction in Robust ASR". In: *INTERSPEECH*.

Marti-Renom, Marc A and Leonid A Mirny (2011). "Bridging the resolution gap in structural modeling of 3D genome organization". In: *PLoS Comput Biol* 7.7, e1002125.

McCulloch, Warren S and Walter Pitts (1943). "A logical calculus of the ideas immanent in nervous activity". In: *The bulletin of mathematical biophysics* 5.4, pp. 115–133.

Meza, Juan C et al. (1996). "A comparison of a direct search method and a genetic algorithm for conformational searching". In: *Journal of Computational Chemistry* 17.9, pp. 1142–1151.

Min, Xu et al. (2016). "DeepEnhancer: Predicting enhancers by convolutional neural networks". In: *Bioinformatics and Biomedicine (BIBM), 2016 IEEE International Conference on*. IEEE, pp. 637–644.

Misteli, Tom (2007). "Beyond the sequence: cellular organization of genome function". In: *Cell* 128.4, pp. 787–800.

Mohamed, Abdel-rahman, George E Dahl, and Geoffrey Hinton (2012). "Acoustic modeling using deep belief networks". In: *IEEE Transactions on Audio, Speech, and Language Processing* 20.1, pp. 14–22.

Mourad, Raphaël and Olivier Cuvier (2015). "Predicting the spatial organization of chromosomes using epigenetic data". In: *Genome biology* 16.1, p. 182.

Nagano, Takashi et al. (2015). "Comparison of Hi-C results using in-solution versus in-nucleus ligation". In: *Genome biology* 16.1, p. 175.

Nair, Vinod and Geoffrey E Hinton (2010). "Rectified linear units improve restricted boltzmann machines". In: *Proceedings of the 27th International Conference on Machine Learning (ICML-10)*, pp. 807–814.

Nora, Elphège P et al. (2012). "Spatial partitioning of the regulatory landscape of the x-inactivation center". In: *Nature* 485.7398, p. 381.

Nowotny, Jackson et al. (2015). "Iterative reconstruction of three-dimensional models of human chromosomes from chromosomal contact data". In: *BMC bioinformatics* 16.1, p. 1.

Ono, Isao et al. (2002). "Global optimization of protein 3-dimensional structures in NMR by a genetic algorithm". In: *Evolutionary Computation, 2002. CEC'02. Proceedings of the 2002 Congress on*. Vol. 1. IEEE, pp. 303–308.

Pancaldi, Vera et al. (2016). "Integrating epigenomic data and 3D genomic structure with a new measure of chromatin assortativity". In: *Genome Biology* 17.1, p. 152.

Pedregosa, F. et al. (2011). "Scikit-learn: Machine Learning in Python". In: *Journal of Machine Learning Research* 12, pp. 2825–2830.

Peng, Cheng et al. (2013). "The sequencing bias relaxed characteristics of Hi-C derived data and implications for chromatin 3D modeling". In: *Nucleic acids research* 41.19, e183–e183.

Poultney, Christopher, Sumit Chopra, Yann L Cun, et al. (2007). "Efficient learning of sparse representations with an energy-based model". In: *Advances in neural information processing systems*, pp. 1137–1144.

Quang, Daniel and Xiaohui Xie (2016). "DanQ: a hybrid convolutional and recurrent deep neural network for quantifying the function of DNA sequences". In: *Nucleic acids research* 44.11, e107–e107.

Rao, Suhas SP et al. (2014). "A 3D map of the human genome at kilobase resolution reveals principles of chromatin looping". In: *Cell* 159.7, pp. 1665–1680.

Rosenblatt, Frank (1958). "The perceptron: A probabilistic model for information storage and organization in the brain." In: *Psychological review* 65.6, p. 386.

Rousseau, Mathieu et al. (2011). "Three-dimensional modeling of chromatin structure from interaction frequency data using Markov chain Monte Carlo sampling". In: *BMC bioinformatics* 12.1, p. 414.

Sanborn, Adrian L et al. (2015). "Chromatin extrusion explains key features of loop and domain formation in wild-type and engineered genomes". In: *Proceedings of the National Academy of Sciences* 112.47, E6456–E6465.

Sandouk, Ubai and Ke Chen (2016). "Learning Contextualized Music Semantics from Tags Via a Siamese Neural Network". In: *ACM Transactions on Intelligent Systems and Technology (TIST)* 8.2, p. 24.

Schmidhuber, Jürgen (2015). "Deep learning in neural networks: An overview". In: *Neural Networks* 61, pp. 85–117.

Schmitt, Anthony D, Ming Hu, and Bing Ren (2016). "Genome-wide mapping and analysis of chromosome architecture". In: *Nature Reviews Molecular Cell Biology*.

Schmitt, Anthony D et al. (2016). "A compendium of chromatin contact maps reveals spatially active regions in the human genome". In: *Cell reports* 17.8, pp. 2042–2059.

Schreiber, Jacob et al. (2017). "Nucleotide sequence and DNaseI sensitivity are predictive of 3D chromatin architecture". In: *bioRxiv*, p. 103614.

Segal, Mark R and Henrik L Bengtsson (2015). "Reconstruction of 3D genome architecture via a two-stage algorithm". In: *BMC bioinformatics* 16.1, p. 373.

Seitan, Vlad C et al. (2013). "Cohesin-based chromatin interactions enable regulated gene expression within preexisting architectural compartments". In: *Genome research* 23.12, pp. 2066–2077.

Sermanet, Pierre et al. (2013). "Pedestrian detection with unsupervised multi-stage feature learning". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3626–3633.

Serre, Thomas et al. (2007). "Robust object recognition with cortex-like mechanisms". In: *IEEE transactions on pattern analysis and machine intelligence* 29.3, pp. 411–426.

Sexton, Tom et al. (2007). "Gene regulation through nuclear organization". In: *Nature structural & molecular biology* 14.11, pp. 1049–1055.

Shavit, Yoli, Fiona Kathryn Hamey, and Pietro Lio (2014). "FisHiCal: an R package for iterative FISH-based calibration of Hi-C data". In: *Bioinformatics* 30.21, pp. 3120–3122.

Shen, Yin et al. (2012). "A map of the cis-regulatory sequences in the mouse genome". In: *Nature* 488.7409, p. 116.

Silver, David et al. (2016). "Mastering the game of Go with deep neural networks and tree search". In: *Nature* 529.7587, pp. 484–489.

Singh, Ritambhara et al. (2016). "DeepChrome: deep-learning for predicting gene expression from histone modifications". In: *Bioinformatics* 32.17, pp. i639–i648.

Sofueva, Sevil et al. (2013). "Cohesin-mediated interactions organize chromosomal domain architecture". In: *The EMBO journal* 32.24, pp. 3119–3129.

Srivastava, Nitish et al. (Jan. 2014). "Dropout: A Simple Way to Prevent Neural Networks from Overfitting". In: *J. Mach. Learn. Res.* 15.1, pp. 1929–1958. ISSN: 1532-4435. URL: http://dl.acm.org/citation.cfm?id=2627435.2670313.

Sukhbaatar, Sainbayar and Rob Fergus (2014). "Learning from noisy labels with deep neural networks". In: *arXiv preprint arXiv:1406.2080* 2.3, p. 4.

Sutskever, Ilya et al. (2013). "On the importance of initialization and momentum in deep learning". In: *International conference on machine learning*, pp. 1139–1147.

Tang, Zhonghui et al. (2015). "CTCF-mediated human 3D genome architecture reveals chromatin topology for transcription". In: *Cell* 163.7, pp. 1611–1627.

Tanizawa, Hideki et al. (2010). "Mapping of long-range associations throughout the fission yeast genome reveals global genome organization linked to transcriptional regulation". In: *Nucleic acids research* 38.22, pp. 8164–8177.

Trieu, Tuan and Jianlin Cheng (2014). "Large-scale reconstruction of 3D structures of human chromosomes from chromosomal contact data". In: *Nucleic acids research* 42.7, e52–e52.

— (2015). "MOGEN: A Tool for Reconstructing 3D Models of Genomes from Chromosomal Conformation Capturing Data". In: *Bioinformatics*, btv754.

Varoquaux, Nelle et al. (2014). "A statistical approach for inferring the 3D structure of the genome". In: *Bioinformatics* 30.12, pp. i26–i33.

Wächter, Andreas and Lorenz T Biegler (2006). "On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming". In: *Mathematical programming* 106.1, pp. 25–57.

Wang, Jiang et al. (2014). "Learning fine-grained image similarity with deep ranking". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1386–1393.

Wang, Siyu, Jinbo Xu, and Jianyang Zeng (2015). "Inferential modeling of 3D chromatin structure". In: *Nucleic acids research*, gkv100.

Weise, Thomas (2009). "Global optimization algorithms-theory and application". In: *Self-Published,*

Williams, DRGHR and Geoffrey Hinton (1986). "Learning representations by back-propagating errors". In: *Nature* 323.6088, pp. 533–538.

Wohlhart, Paul and Vincent Lepetit (2015). "Learning descriptors for object recognition and 3d pose estimation". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3109–3118.

Yaffe, Eitan and Amos Tanay (2011). "Probabilistic modeling of Hi-C contact maps eliminates systematic biases to characterize global chromosomal architecture". In: *Nature genetics* 43.11, pp. 1059–1065.

Yang, Jianchao et al. (2008). "Image super-resolution as sparse representation of raw image patches". In: *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*. IEEE, pp. 1–8.

Yang, Tao et al. (2017). "HiCRep: assessing the reproducibility of Hi-C data using a stratum-adjusted correlation coefficient". In: *bioRxiv*, p. 101386.

Zeiler, Matthew D and Rob Fergus (2014). "Visualizing and understanding convolutional networks". In: *European Conference on Computer Vision*. Springer, pp. 818–833.

Zeng, Haoyang et al. (2016). "Convolutional neural network architectures for predicting DNA–protein binding". In: *Bioinformatics* 32.12, pp. i121–i127.

Zhang, Cheng et al. (2016). "Siamese neural network based gait recognition for human identification". In: *Acoustics, Speech and Signal Processing (ICASSP), 2016 IEEE International Conference on*. IEEE, pp. 2832–2836.

Zhang, Yan et al. (2017). "HiCPlus: Resolution Enhancement of Hi-C interaction heatmap". In: *bioRxiv*, p. 112631.

Zhang, Yao-zhong et al. (2017). "Sequence-specific bias correction for RNA-seq data using recurrent neural networks". In: *BMC genomics* 18.1, p. 1044.

Zhang, ZhiZhuo et al. (2013). "3D chromosome modeling with semi-definite programming and Hi-C data". In: *Journal of computational biology* 20.11, pp. 831–846.

Zheng, Lilei et al. (2016). "Siamese multi-layer perceptrons for dimensionality reduction and face identification". In: *Multimedia Tools and Applications* 75.9, pp. 5055–5073.

Zhou, Jian and Olga G Troyanskaya (2015). "Predicting effects of noncoding variants with deep learning–based sequence model". In: *Nature methods* 12.10, p. 931.

Zhou, Jiyun et al. (2016). "CNNsite: Prediction of DNA-binding residues in proteins using Convolutional Neural Network with sequence features". In: *Bioinformatics and Biomedicine (BIBM), 2016 IEEE International Conference on*. IEEE, pp. 78–85.

Zhu, Yun et al. (2016). "Constructing 3D interaction maps from 1D epigenomes". In: *Nature communications* 7, p. 10812.